

#4
10/18/02

Docket No.: 50006-140

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of :
Tomohiro MORIMURA, et al. :
Serial No.: : Group Art Unit:
Filed: March 01, 2002 : Examiner:
For: MULTI-PROCESSOR SYSTEM APPARATUS

1017 U.S. PTO
10/085132
03/01/02

**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Commissioner for Patents
Washington, DC 20231

Sir:

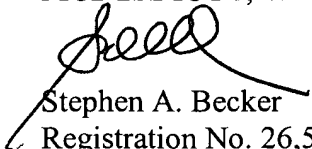
In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 2001-056475, filed March 1, 2001

A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Stephen A. Becker
Registration No. 26,527

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 SAB:mlw
Date: March 1, 2002
Facsimile: (202) 756-8087

534472 (45)

50006-140

Tomohiro MORIMURA et.

日 本 国 特 許 庁 March 1, 2002

JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 3月 1日

出 願 番 号

Application Number:

特願2001-056475

[ST.10/C]:

[JP 2001-056475]

出 願 人

Applicant(s):

株式会社半導体理工学研究センター

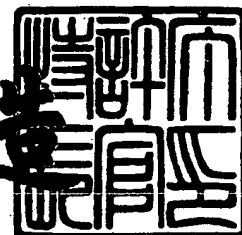
J1017 U.S. PTO
10/085132
03/01/02

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 1月25日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願

【整理番号】 172781

【提出日】 平成13年 3月 1日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16

【発明者】

 【住所又は居所】 神奈川県横浜市港北区日吉 3 - 6 - 8 グランブル日吉
 1 0 1 号室

 【氏名】 森村 知弘

【発明者】

 【住所又は居所】 神奈川県横浜市港北区日吉 7 - 1 6 - 2 7 - 1 0 3

 【氏名】 天野 英晴

【特許出願人】

 【識別番号】 396023993

 【住所又は居所】 東京都港区新橋 6 丁目 1 6 番 1 0 号

 【氏名又は名称】 株式会社半導体理工学研究センター

【代理人】

 【識別番号】 100062144

 【弁理士】

 【氏名又は名称】 青山 葆

【選任した代理人】

 【識別番号】 100086405

 【弁理士】

 【氏名又は名称】 河宮 治

【手数料の表示】

 【予納台帳番号】 013262

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9608010

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチプロセッサシステム装置

【特許請求の範囲】

【請求項 1】 複数のプロセッサがネットワークを介して相互に接続されてなるマルチプロセッサシステム装置において、

プロセッサ、メモリ部及び上記ネットワークとのインタフェースを行うインタフェース部からなる複数のプロセッサエレメントと、

該各プロセッサエレメント間の接続を行う多段のスイッチによって構成された、階層構造を有する多段結合網と、
を備え、

上記各プロセッサエレメント及び多段結合網は、所定の数を経数とした階層構造にクラスタリングされると共に、各時刻ごとに生成された上記多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット転送を行うことを特徴とするマルチプロセッサシステム装置。

【請求項 2】 上記階層構造を有する多段結合網は、下位階層から上位階層にパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層にパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えることを特徴とする請求項 1 記載のマルチプロセッサシステム装置。

【請求項 3】 上記スイッチ状態表は、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成されることを特徴とする請求項 1 又は 2 記載のマルチプロセッサシステム装置。

【請求項 4】 上記各プロセッサエレメント及び多段結合網は、1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うことを特徴とする請求項 1、2 又は 3 記載のマルチプロセッサシステム装置。

【請求項 5】 上記多段結合網は、クロス網であり、上記各プロセッサエレメント及び多段結合網は、1つのクロス網内のパケット転送時に、1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うことを特徴とする請求項 4 記載のマルチプロセッサシステム装置。

【請求項 6】 上記各パケットに対するスケジューリングは、コンパイラによってあらかじめ行われることを特徴とする請求項 1、2、3、4 又は 5 記載のマルチプロセッサシステム装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数のプロセッサを使用したマルチプロセッサシステム装置に関し、特に、多数のプロセッサとメモリモジュールを多段のスイッチ（多段結合網）によって接続する構成をなすマルチプロセッサシステム装置に関する。

【0002】

【従来の技術】

多数のプロセッサとメモリモジュールをスイッチによって接続する構成を有するマルチプロセッサシステム装置において、1つのスイッチに複数のパケットが集中して衝突した場合、データ処理に時間を要しデータ処理性能が低下するという問題があった。このため、スイッチにおけるパケットの衝突を低減することができるノンブロッキング網、リアレンジブル網及びブロッキング網が提案されていた。

【0003】

ノンブロッキング網は、クロスバ網やクロス(Clos)網等があり、スケジューリングによって出線競合を回避すればスイッチ内では衝突が起きることはない。また、リアレンジブル網は、それぞれのスイッチ構成要素の設定をスケジュールすることによって衝突をなくすることができる。一方、ブロッキング網は、一般的に

はスケジューリングによって衝突をなくすることができないが、一定のアクセスパターンに対してはスケジューリングによって衝突をなくすることができる。

【0004】

【発明が解決しようとする課題】

しかし、ノンブロッキング網では、プロセッサ及びメモリモジュールの数に対するハードウェア量の増加が大きいため、大規模なシステムに使用するとコストが膨大なものとなる。また、リアレンジブル網は、ノンブロッキング網と比較してハードウェアのコストは小さいが、スケジューリングに要する時間が多大なものとなり、マルチプロセッサでの利用は困難であった。更に、従来のブロッキング網のスケジューリングは、一定のアクセスパターンの並べ換えに対してのみ無衝突にする方法であり、マルチプロセッサで実際に利用することができるのは、配列要素が一定の順番に並んでいる特殊な場合に制限されていた。

【0005】

本発明は、上記のような問題を解決するためになされたものであり、多数のマルチプロセッサとメモリモジュールを使用した大規模なシステムに対して、コンパイラが容易に静的スケジューリングを行うことができ、一般的な同時アクセスパターンに対して無衝突なパケット転送を実現することができるマルチプロセッサシステム装置を得ることを目的とする。

【0006】

【課題を解決するための手段】

この発明に係るマルチプロセッサシステム装置は、複数のプロセッサが所定のネットワークを介して相互に接続されてなるマルチプロセッサシステム装置において、プロセッサ、メモリ部及びネットワークとのインタフェースを行うインタフェース部からなる複数のプロセッサエレメントと、該各プロセッサエレメント間の接続を行う多段のスイッチによって構成された、階層構造を有する多段結合網とを備え、各プロセッサエレメント及び多段結合網は、所定の数をもとにした階層構造にクラスタリングされると共に、各時刻ごとに生成された多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット

転送を行うものである。

【0007】

また、上記階層構造を有する多段結合網は、下位階層から上位階層にパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層にパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えるようにしてもよい。

【0008】

具体的には、上記スイッチ状態表は、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成されるようにした。

【0009】

また、上記各プロセッサエレメント及び多段結合網は、1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにした。

【0010】

一方、上記多段結合網がクロス網の場合、各プロセッサエレメント及び多段結合網は、1つのクロス網内のパケット転送時に、1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにしてもよい。

【0011】

具体的には、上記各パケットに対するスケジューリングを、コンパイラによってあらかじめ行うようにした。

【0012】

【発明の実施の形態】

次に、図面に示す実施の形態に基づいて、本発明を詳細に説明する。

第1の実施の形態。

図1は、本発明の第1の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図である。

図1において、マルチプロセッサシステム装置1は、数十から数千程度のプロセッサエレメントPEを、階層構造をもつ多段結合網(Multistage Interconnection Network:MIN)で接続してなる。なお、図1では、3階層の場合を例にして示している。

【0013】

マルチプロセッサシステム装置1は、クラスタD0～Dx (xは、 $x > 0$ の整数)、及び該各クラスタD0～Dx間の接続を行う結合網(Interconnection Network)E0で構成されている。更に、各クラスタD0～Dxは、クラスタA0～An (nは、 $n > 0$ の整数)、及び該各クラスタA0～An間の接続を行う結合網C0～Cxでそれぞれ構成されている。また、各クラスタA0～Anは、プロセッサエレメントPE0～PEm (mは、 $m > 0$ の整数)、及び該各プロセッサエレメントPE0～PEm間の接続を行う結合網B0～Bnでそれぞれ構成されている。

【0014】

すなわち、マルチプロセッサシステム装置1では、数百～数千のプロセッサエレメントPEを結合するためにネットワークを階層構造に分離し、十数～数十のプロセッサエレメントPEを、中規模なマルチプロセッサシステム装置に採用される多段結合網という複数のスイッチを数段にわたって接続し、目的のプロセッサエレメントPEへは、途中のスイッチの切り換えによって転送経路が形成されている。

【0015】

プロセッサエレメントPE0～PEmはそれぞれ同じ構成であることからプロセッサエレメントPEi ($i = 0 \sim m$)を例にして説明する。

図2は、プロセッサエレメントPEiの構成例を示した概略のブロック図である。

図2において、プロセッサエレメントPEiは、プロセッサPU、メモリME及びネットワークインタフェースNIで構成されている。プロセッサPUとメモ

リMEは接続され、更にプロセッサPU及びメモリMEはネットワークインタフェースNIを介して対応する結合網Biに接続されている。

【0016】

このような構成において、各クラスタA0～Anにおける同一クラスタ内のプロセッサエレメント間の接続を行う構成をレベル0とし、各クラスタA0～An間での接続を行う構成をレベル1とし、各クラスタD0～Dx間での接続を行う構成をレベル2とする。すなわち、各クラスタA0～Anがレベル0であり、各クラスタD0～Dxがレベル1であり、結合網E0がレベル2となり、レベル0～2の3階層を形成している。言い換えれば、各クラスタD0～Dx及び結合網E0はクラスタF0とすることができ、クラスタF0がレベル2となる。

【0017】

ここで、一般的な多段結合網の1つであるクロス(Clos)網の例を図3に示す。

クロス網では、各段のスイッチの役割から1段目をディストリビュータ(distributor)、2段目をエクスチェンジャ(exchanger)、3段目をコンセントレータ(concentrator)と呼ぶ。なお、図3では、4入力4出力のスイッチを使用し、4つの該スイッチで各段を構成した場合を例にして示している。

【0018】

多段結合網は、結合しているノードの数、すなわちマルチプロセッサシステム装置ではプロセッサエレメントPEの数(m+1)と構成要素のスイッチにおける入力端子又は出力端子の数kによって、すべてのプロセッサエレメントPEに対して転送経路が形成可能となる段数は、 $\log_k(m+1)$ となる。図3では、相対するプロセッサエレメントPEは、同じプロセッサエレメントを示していることから $m+1=16$ となり $k=4$ である。

【0019】

このため、スイッチを2段介することによってすべてのプロセッサエレメントPEへの転送経路が形成されるが、より大きな転送容量を得ると共に転送経路に冗長性を持たせるために3段のスイッチによってクロス網が構成されている。すなわち、1つのプロセッサエレメントPEには、ディストリビュータをなすスイッチエレメントの1つの入力端子とコンセントレータをなすスイッチエレメント

の 1 つの出力端子が対応して接続されている。

【 0 0 2 0 】

このような多段結合網は、構成スイッチの入出力数と段数及び構成スイッチの数によって、ノンブロッキング、リアレンジブル及びブロッキングの 3 通りに分類することができる。ノンブロッキングは、静的に、転送データの衝突を起こさない転送経路を設定することができ、リアレンジブルは、転送データの衝突が発生した場合に、転送経路を再設定することによって無衝突な経路を形成することができる。ブロッキングは、転送データの衝突が発生した場合に、転送経路を再設定しても無衝突な経路を形成することができない。例えば、図 3 で示したクロス網では、構成スイッチの入力端子数又は出力端子数を k とし、中間段のスイッチ数を p とすると、 $p > (2k - 1)$ のときはノンブロッキング、 $p \geq k$ のときはリアレンジブル、 $p < k$ のときはブロッキングとなる。

【 0 0 2 1 】

一方、多段結合網で数百～数千ものプロセッサエレメント PE を接続することは、ハードウェア的に困難で現実的ではない。このため、数個のプロセッサエレメント PE をクロスバ (Crossbar) スwitch に接続してレベル 0 のネットワークとすると共に、該クロスバ switch を入力とする多段結合網で十数～数十のプロセッサエレメント PE を結合してレベル 1 のネットワークとする。更に、複数の該多段結合網を結合するための拡張段を複数の switch で形成してレベル 2 のネットワークとする。

【 0 0 2 2 】

同様に、システム装置の規模に応じた階層の拡張段を付加することによって大規模なシステム装置を相互結合させて、多段結合網を基本網とした階層構造に拡張させることにより、スケーラビリティを得ることができる。このように、各階層のネットワークは、1 つのサブネットワークとしてとらえることが可能であるため、各階層レベルに応じてレベル s (s は、 $s > 0$ の整数) のネットワーク NW_s と呼ぶ。

【 0 0 2 3 】

このような実例として、クロス網を基本多段結合網とした階層構造ネットワー

クについて説明する。

図4及び図5は、基本網となるクロス網内における階層サブネットワークの例を示した図であり、図4は、クロス網内におけるレベル0のネットワークの例を、図5は、クロス網内におけるレベル1のネットワークの例を示している。なお、図4及び図5では、4つのプロセッサエレメントPE0～PE3を有する4つのクラスタA0～A3を例にして示している。

【0024】

図4及び図5において、スイッチエレメントSD0～SD3は、クロス網のディストリビュータをなし、スイッチエレメントSE0～SE3は、クロス網のエクステンジャをなし、スイッチエレメントSC0～SC3は、クロス網のコンセントレータをなしている。また、スイッチエレメントSD0～SD3、SE0～SE3、SC0～SC3は、それぞれ4入力4出力のスイッチエレメントをなしている。

【0025】

スイッチエレメントSD0及びSC0は、接続された各プロセッサエレメントPE0～PE3と共にクラスタA0を形成し、スイッチエレメントSD1及びSC1は、接続された各プロセッサエレメントPE0～PE3と共にクラスタA1を形成している。同様に、スイッチエレメントSD2及びSC2は、接続された各プロセッサエレメントPE0～PE3と共にクラスタA2を形成し、スイッチエレメントSD3及びSC3は、接続された各プロセッサエレメントPE0～PE3と共にクラスタA3を形成している。

【0026】

結合網C0をなすスイッチングエレメントSE0～SE3が、例えば図4の矢印で示すように入力端子と同じ出力端子へストレートにスイッチングされると、レベル0のネットワークが形成され、クラスタA0～A3における同一クラスタ内でのデータ転送が実現する。これに対して、結合網C0をなすスイッチングエレメントSE0～SE3が、例えば図5の矢印で示すように入力端子が異なる出力端子にクロスしてスイッチングされると、エクステンジャをなす2段目のスイッチエレメントSE0～SE3が、レベル1のネットワークをなし、クラスタ

A0～A3における異なるクラスタ間でのデータ転送が実現する。

【0027】

このように、エクステンジャをなす2段目のスイッチエレメントSE0～SE3は、スイッチとしての役割を果たした場合、レベル1のネットワークとして動作し、スイッチとしての役割を果たさなかった場合は、レベル0のネットワークとして動作したことになる。すなわち、1つのクロス網内には、レベル0のネットワークとレベル1のネットワークという2つのサブネットワークが存在することになる。

【0028】

次に、クロス網間を接続する拡張段、すなわち図1の結合網E0について説明する。

図6は、階層構造クラスタリングを実現したマルチプロセッサシステム装置1の例を示した図である。なお、図6では、説明を分かりやすくするために、4を基数、すなわち4つのプロセッサエレメントPE0～PE3を有する4つのクラスタA0～A3を備えた4つのクラスタD0～D3で構成される階層構造クラスタリングを実現した場合を例にして示し、プロセッサエレメントは省略して示している。

【0029】

図6において、16個のプロセッサエレメントまでをクロス網で直接結合してクラスタD0～D3をそれぞれ形成し、該クロス網同士、すなわちクラスタD0～D3を、付加した拡張段であるレベル2のネットワークにおけるエクステンジャをなすスイッチエレメントSEa0～SEa3を使用して相互結合を行う。該スイッチエレメントSEa0～SEa3は、それぞれ4入力4出力のスイッチエレメントをなし、図1の結合網E0をなす。またこの場合、各クロス網内のスイッチエレメントSE0～SE3は、スイッチエレメントSEa0～SEa3に接続するための1入力1出力が追加され、5入力5出力となる。

【0030】

また、更に多くのプロセッサエレメントを結合させる場合は、レベル2のネットワークを相互結合させるためのレベル3のネットワークとしてレベル3のエク

スチェンジャを付加する。すなわち、図1で示したすべての構成を有するクラスタが複数存在し、該各クラスタ間の結合を行う結合網を設けて4階層にする。このように、階層数Rである場合、結合されるプロセッサエレメントの数Nは、基本網である多段結合網に接続されるプロセッサエレメント数(m+1)から、下記(1)式のようになる。

$$N = (m+1) \times k \quad (R-1) \dots\dots\dots (1)$$

【0031】

また、図6のマルチプロセッサシステム装置1において、 $m+1 = k \times k$ となることから、上記(1)式は下記(2)式のようになる。

$$N = k \times k \times k \quad (R-1) = k \quad (R+1) \dots\dots\dots (2)$$

【0032】

次に、上記のような階層構造を有する多段結合網の静的スケジューリング方法について説明する。

階層構造を有する多段結合網において静的スケジューリングする場合の前提条件として、すべてのデータ転送は、コンパイラにおけるスケジューラによって完全に静的解析がなされ、どのタイミングでどこあてのパケットが転送されるという情報が分かっている上でデータアクセスをスケジューリングするものとする。

【0033】

静的にスケジューリングするためには各時刻におけるスイッチの状態を把握しなければならず、各スイッチエレメントの出力端子ごとに、「現在時刻」、「保持ポート」、「保持クロック」、「ポート要求待ち行列」及び「状態」といった各項目から成り立つスイッチ状態表を作成する。なお、「保持ポート」とは、この出力端子を保持している入力ポート番号であり、「保持クロック」とは、保持しているサイクル(クロック)数である。また、「ポート要求待ち行列」とは、この出力端子を要求している入力端子番号を入れる待ち行列であり、「状態」とは、この出力端子の状態を示しリリース(RELEASED)とホールド(HOLD)の2つの状態がある。

【0034】

図7は、スイッチ状態表の例を示した図である。なお、図7では、4入力4出

力のスイッチエレメントの場合を例にして示している。

図7において、現在時刻が157843のときにおける各スイッチの状態を示しており、出力端子#0が入力端子#3によって2クロック保持されている。したがって、この2クロックの間は、他の入力端子の packets が出力端子#0を獲得することはできない。また、出力端子#1は開放されているが、入力端子#0と#2の各 packets が出力端子#1の獲得要求を出している。出力端子#2及び#3においても開放されているが、入力端子#1の packets が出力端子#2の獲得要求を出している。

【0035】

各スイッチエレメントにおけるすべてのスイッチにおいて図7で示したようなスイッチ状態表が作成され、コンパイラにおけるスケジューラは該スイッチ状態表に基づいてスケジューリングを行う。図7の出力端子#1のように、ポート要求待ち行列に2つ以上の獲得要求がある場合は、packetsの優先度に基づいて調停され、調停に負けたpacketsのアクセスは後の時刻にずらされる。

【0036】

一方、調停に勝ったpacketsは、出力端子を獲得して保持ポートと保持クロックに記載され、保持クロックが1となる時刻まで保持ポートに記載されると共に状態をホールドにしてスイッチ状態表が作成される。このようなことから、最終的にはすべてのアクセス時間分のスイッチ状態表が必要になるが、ある時刻のアクセスのpacketsをスケジュールするときに必要なスイッチ状態表は、該時刻よりも後のものだけであるため、該時刻よりも前の時刻のスイッチ状態表は破棄することができる。

【0037】

次に、コンパイラによって行われる、スイッチ状態表を使用した静的スケジューリング方法について説明する。なお、ある時刻Tsに発行されるpacketsの集合Utsを、 $Uts = p_0, p_1, \dots, p_N$ と表すこととし、以下静的スケジューリング方法の各処理において、特に明記しない場合はすべてコンパイラによって行われるものである。

【0038】

まず、パケット集合 U_{ts} の要素であるパケット p_j ($j = 0 \sim N$) に対して、対応するディストリビュータのスイッチエレメントにおけるスイッチ状態表を、パケットのヘッダ(ルーティングタグ等)に応じて作成する。また、スイッチ状態表の現在時刻を T_s に設定する。次に、すべてのパケット $p_1 \sim p_N$ に対して、ディストリビュータのスイッチエレメントにおけるスイッチ状態表が作成されると、スイッチ状態表におけるポート要求待ち行列内の入力端子のパケットに対して調停を行う。調停に負けたパケットは、パケット集合 U_{ts} から除かれて次の時刻 T_{s+1} に発行されるパケット集合 U_{ts+1} に加えられる。

【 0 0 3 9 】

一方、調停に勝ったパケットに対しては、獲得した出力端子のスイッチ状態表を、保持クロック数分だけ作成又は書き換えを行う。各スイッチエレメントにおけるすべてのスイッチに対する状態が決定すると、出力に応じて対応する次の段のスイッチに対するスイッチ状態表を作成又は書き換えを行う。このときのスイッチ状態表は、現在時刻を 1 つ進めたときのスイッチ状態を示す。

このような処理を繰り返し、目的地に到着したパケットはその都度パケット集合 U_{ts} から取り除かれ、パケット集合 U_{ts} が空集合になるまでこのような処理を繰り返す。

【 0 0 4 0 】

上記のような操作によってある時刻 T_s に発行されたパケットは、無衝突に調整及びスケジューリングされる。また、上記の操作で、パケット集合内に同じノードから発行されるパケットが 2 つある場合、その内 1 つは次の時刻のパケット集合に入れられるため、パケットのアクセスが集中している場合は、1 つずつパケット集合がずれていくことになる。該スケジューリング対象の時刻 T_s と同様の処理を全時刻に対して行うことによって静的にパケット転送を完全にスケジューリングすることができる。

【 0 0 4 1 】

図 8 ～ 図 1 0 は、スイッチ状態表を使用した静的なスケジューリング方法を示したフローチャートであり、図 8 ～ 図 1 0 を用いて静的なスケジューリング処理の流れについてもう少し詳細に説明する。なお、図 8 ～ 図 1 0 では、ある時刻 T

sに発行されるパケットの集合 U_{ts} を、 $U_{ts} = p_0, p_1, \dots, p_N$ と表すこととする。また、図8～図10の各フローで行われる処理は、特に明記しない限りコンパイラによって行われる。

【0042】

図8において、まず最初に時刻 T_s に発行されるパケット集合 U_{ts} を1段目の各スイッチの入力端子にエントリする（ステップS1）。なお、多段結合網の段数は、入力側から数字を1から昇順に振るものとする。次に、現在注目しているスイッチの段数 S_{Tcur} を1に設定すると共に、現在処理している最高位の階層数 R_{cur} を1に設定する（ステップS2）。この後、段数 S_{Tcur} の各スイッチに対してスケジューリングを行う（ステップS3）。段数 S_{Tcur} の各スイッチに下位階層へのリンクが存在しているか否かを調べ（ステップS4）、存在している場合は（YES）、段数 S_{Tcur} を下位階層の段数に設定し現在時刻 T_{cur} を1つ進め（ステップS5）、この後ステップS3に戻る。

【0043】

一方、ステップS4で、下位階層へのリンクが存在していなかった場合（NO）、段数 S_{Tcur} の各スイッチに上位階層へのリンクが存在しているか否かを調べ（ステップS6）、存在している場合は（YES）、現在処理している最高位の階層数 R_{cur} を1増やすと共に現在注目しているスイッチの段数 S_{Tcur} を該階層数 R_{cur} と同じ数に設定した（ステップS7）後、ステップS3に戻る。また、ステップS6で、上位階層へのリンクが存在していなかった場合（NO）、本フローは終了する。

【0044】

ここで、図8のステップS3で示したスケジューリング処理について、図9のフローチャートを用いてもう少し詳細に説明する。

図9において、最初に、段数 S_{Tcur} に属するすべてのスイッチエレメントに対して、入力端子にエントリされたパケットの行き先出力端子番号に基づいて、対応する時刻 T_{cur} のスイッチ状態表のポート要求待ち行列に入力端子番号をエントリする（ステップS11）。次に、段数 S_{Tcur} に属する各スイッチエレメントに対して順に0から番号を振り、注目しているスイッチエレメント番号 SW

curを0に設定する（ステップS12）。

【0045】

この後、SWcurのスイッチエレメントに対して、スイッチ状態表によるスケジューリングを行い（ステップS13）、現在注目している段数STcurが最終段であるか否かを調べる（ステップS14）。ステップS14で、最終段である場合（YES）、出力端子にエントリされたパケットを到着パケットとしてパケット集合Utsから削除する（ステップS15）。更に、注目しているスイッチエレメント番号SWcurを1つ進め（ステップS16）、該スイッチエレメント番号SWcurがその段数STcurの全スイッチ数Nst未満であるか否かを調べる（ステップS17）。

【0046】

ステップS17で、全スイッチ数Nst未満である場合（YES）、本フローは終了して図8のステップS4に進む。また、ステップS17で、全スイッチ数Nst未満でない場合（NO）、ステップS13に戻る。また、ステップS14で、最終段でなかった場合（NO）、出力端子にエントリされたパケットを接続されている次の段のスイッチの入力端子にエントリさせ（ステップS18）、ステップS16に進む。

【0047】

ここで、図9のステップS13で示したスケジューリング処理について、図10のフローチャートを用いてもう少し詳細に説明する。

図10において、スイッチ状態表の注目している出力端子番号POcurを0に設定し（ステップS21）、該出力端子番号POcurにポート要求待ち行列があるか否かを調べる（ステップS22）。ステップS22で、ポート要求待ち行列がある場合は（YES）、パケットのヘッダ内の優先度に基づいて調停を行い（ステップS23）、ポート要求待ち行列から1つのパケットを選択し（ステップS24）、該パケットが調停に勝ったか否かを調べる（ステップS25）。

【0048】

ステップS25で、選択したパケットが調停に勝つと（YES）、該パケットを書き込むスイッチ状態表の時刻Thを時刻Tcurに設定する（ステップS26）

。この後、該パケットの入力端子番号を、時刻 T_h におけるスイッチ状態表の獲得した出力端子番号の保持ポートに書き込み（ステップ S_{27} ）、該パケットが通過するのに要するクロック数を保持クロックに書き込む（ステップ S_{28} ）。次に、保持クロックに書き込まれたクロック数を1つ減らし、現在時刻 T_h を1つ進め（ステップ S_{29} ）、保持クロックに書き込まれたクロック数が0でないか否かを調べる（ステップ S_{30} ）。ステップ S_{30} で、0でない場合は（YES）、ステップ S_{27} に戻り、0の場合は（NO）、ステップ S_{22} に戻る。

【0049】

一方、ステップ S_{25} で、取り出したパケットが調停に負けると（NO）、負けたパケットをパケット集合 U_{ts} から取り除き、次の時刻のパケット集合 U_{ts+1} に加え、同じノードの後続パケットの発行の重複がなくなるまで1つずつパケット集合をずらし（ステップ S_{31} ）、ステップ S_{22} に戻る。また、ステップ S_{22} で、ポート要求待ち行列がない場合は（NO）、出力端子番号 P_{Ocur} を1つ進め（ステップ S_{32} ）、出力端子番号 P_{Ocur} がスイッチの出力端子数 N_{port} 未満であるか否かを調べる（ステップ S_{33} ）。ステップ S_{33} で、出力端子数 N_{port} 未満である場合（YES）、ステップ S_{22} に戻り、出力端子数 N_{port} 未満でない場合（NO）は、本フローは終了して図9のステップ S_{14} に進む。

【0050】

上記のようなスケジューリング方法に対して、具体的な例を示しながら説明する。例えば、アクセス発行時刻 T_s のパケットのスケジューリングにおいて、時刻15000のある階層ネットワークに属するスイッチが図11で示したような状況である場合について説明する。なお、図11では、エクステンジャを構成する5入力5出力のスイッチエレメントを例にして示している。

図11において、出力端子#2は、入力端子#4によって2クロックだけ保持されている。入力端子のパケットは、ルーティングタグによって適切な出力端子のポート要求待ち行列に入る。図11のスイッチ状態表は、スケジューリングが行われる前の状態を示しており、コンパイラは、図11のスイッチ状態表を基にして調停を行い、図12で示したスイッチ状態表を作成する。

【0051】

図 1 1 において、出線競合があるのは出力端子 # 1 であり、この場合、コンパイラは、パケットのヘッダ内の優先度に応じて調停を行う。仮に入力端子 # 1 のパケットが調停に勝ったとすると、調停に負けた入力端子 # 0 のパケットは、コンパイラによって、アクセス発行時刻 T_s のパケット集合 U_{ts} の要素から取り除かれ、次の発行時刻のパケット集合 U_{ts+1} に加えられ、図 1 2 のように入力端子 # 0 のパケットが出力端子 # 1 に対するエントリから外される。調停に勝った入力端子 # 1 のパケットは、コンパイラによって、図 1 2 のように出力端子 # 1 の保持ポートに入れられると共に出力端子 # 1 の保持クロックに 1 が書き込まれ、更に出力端子 # 1 の状態がホールドに設定される。

【 0 0 5 2 】

次に、出力端子 # 2 においては、図 1 1 のようにすでに状態がホールドに設定され入力端子 # 4 のパケットによって 2 クロック保持されている。このため、出力端子 # 2 を要求している入力端子 # 3 のパケットは、調停に負けたパケットと同様に、コンパイラによって、パケット集合 U_{ts} の要素から取り除かれて、次の発行時刻のパケット集合 U_{ts+1} に加えられ、図 1 2 のように入力端子 # 3 のパケットが出力端子 # 2 に対するエントリから外される。出力端子 # 4 においては、図 1 1 のように状態がリリースに設定されており競合する入力端子のパケットもないことから、入力端子 # 2 のパケットは、コンパイラによって、図 1 2 のように出力端子 # 4 の保持ポートに入れられると共に出力端子 # 4 の保持クロックに 1 が書き込まれ、更に出力端子 # 4 の状態がホールドに設定される。

【 0 0 5 3 】

このように、調停が完了して図 1 2 のスイッチ状態表が作成されると、コンパイラによって、現在時刻が 1 つ進められ、出力端子を獲得したパケットは、該出力端子に接続されるスイッチの入力端子にエントリされ、出力端子 # 4 のパケットは、上位階層のスイッチの入力端子にエントリされる。該エントリされたパケットは、コンパイラによって、上記と同様にポート要求待ち行列にエントリし、調停及び出力端子の獲得操作を目的地に到着するまで繰り返される。なお、上記説明では、階層構造化する多段結合網としてクロス網を使用した場合を例にして示した。しかし、本発明は、これに限定するものではなく、オメガ (Ω) 網、

ベースライン(Baseline)網、デルタ(Delta)網及び「Generalized Cube」等の一般的な多段結合網を使用して階層構造化を行うようにしても実現することができる。

【 0 0 5 4 】

ここで、上記スケジューリング方法では、調停に負けたパケットは次以降の時刻のパケット集合に加えられるようにした。これに対して、1つのクロス網内でパケット転送を行う場合は、レベル1のエキスチェンジャのスイッチエレメントに対するスケジューリングを行った際、調停に負けたパケットをレベル1のエキスチェンジャの他のスイッチエレメントにおける空きのスイッチを使用して転送するようにしてもよい。このようにする場合のスケジューリング方法について、図13の1つのクロス網、すなわちクラスタD0を例にして説明する。

【 0 0 5 5 】

クロス網の性質上、クロス網内の目的地への経路は、2段目のエキスチェンジャと3段目のコンセントレータによって決定されることから、1段目のディストリビュータは任意の出力を選んでよい。クロス網の転送性能は、2段目のエキスチェンジャのスケジューリング性能に大きく依存するため、2段目のエキスチェンジャのスケジューリング結果に応じた出力端子に転送する。したがって、2段目のエキスチェンジャのスケジューリングを先に行ってから、1段目のディストリビュータのスケジューリングを行う。

【 0 0 5 6 】

クロス網の転送性能は、2段目のエキスチェンジャのスケジューリング性能に左右されるため、非常に重要である。2段目のエキスチェンジャを効率よく生かすために上記のようなスイッチ状態表の他に、クラスタ別アクセスリストAL(Access List)とクラスタ別空きポートカウンタVPC(Varid Port Counter)をスケジューリングで使用する。クラスタ別アクセスリスト(以下、アクセスリストと呼ぶ)ALとは、各レベル0のクラスタから出力されたパケットが、どのレベル0のクラスタに向かっているかを記録したリストであり、クラスタ別空きポートカウンタ(以下、空きポートカウンタと呼ぶ)VPCは、各レベル0のクラスタごとに、接続されている出力端子がいくつあるかを表すカウンタである。

【0057】

ここで、コンパイラによって行われるアクセスリストAL及び空きポートカウンタVPCの作成方法について、図13を用いて説明する。なお、図13においても、4を基数、すなわち4つのプロセッサエレメントPE0～PE3を有する4つのクラスタA0～A3を備える4つのクラスタD0～D3で構成された階層構造クラスタリングを実現した場合を例にして示している。クラスタD0は、クラスタA0～A3及びエクスチェンジャをなすスイッチエレメントSE0～SE3で構成されている。更に、クラスタA0～A3は、対応するスイッチエレメントSD0～SD3及びSC0～SC3とプロセッサエレメントPE0～PE3からそれぞれ形成されている。

【0058】

このような構成において、まず、アクセスリストALの作成方法について説明する。

コンパイラは、スイッチエレメントSD0からスイッチエレメントSE0～SE3に転送されたすべてのパケットのヘッダを調べ、該各パケットにおけるあて先のクラスタ番号をアクセスリストALにそれぞれ書き込む。例えば、スイッチエレメントSD0からの各パケットは、スイッチエレメントSE0～SE3のルーティングタグとして、クラスタA1及びA3の2つの要素を有していた場合、アクセスリストALにおけるクラスタA0には、クラスタ番号A1及びA3が書き込まれる。

【0059】

次に、空きポートカウンタVPCの作成方法について説明する。

コンパイラは、作成したアクセスリストALに基づいて、クラスタA0～A3ごとに、割り当てられる出力端子の空きがいくつあるかを表すカウント値CT0～CT3を下記(3)式から対応させて算出する。

$$CT_g = (2 \text{ 段目のスイッチ数}) - (\text{クラスタ別アクセスリストの要素数}) \cdots \cdots (3)$$

なお、上記(3)式において、 $g = 0 \sim 3$ である。

例えば、クラスタA0に対するカウント値CT0は、 $CT0 = 4 - 2 = 2$ とな

る。

【 0 0 6 0 】

次に、アクセスリスト A L 及び空きポートカウンタ V P C を用いてコンパイラによって行われるスケジューリングアルゴリズムについて説明する。ただし、現在のアクセスリストを A L cur とし、調停後のアクセスリストを A L new とする。

まず、コンパイラは、現在のアクセスリスト A L cur における要素数が最も少ないクラスタのパケットから順に、スイッチエレメント S E 0 から順に優先的に割り当てていく。次に、コンパイラは、アクセスリスト A L cur における 1 つのクラスタの要素（あて先クラスタ番号）において、パケットの送り主とあて先が同じクラスタである要素の優先度を最も低くし、同じでない場合は、例えばあて先のクラスタ番号の小さい方から順に割り当てる。なお、あて先のクラスタ番号の大きい方から順に割り当てるようにしてもよい。

【 0 0 6 1 】

また、コンパイラは、アクセスリスト A L cur の要素に対応する空きポートカウンタが 0 である場合、必然的にスケジューリング不能となり、該要素を現在の時刻のパケット集合から取り除き、次の時刻のパケット集合に加える。このようなパケットの割り当てで競合が発生した場合、コンパイラは、パケットの優先度又はラウンドロビン方式等で調停を行う。該調停に勝ったパケットは、コンパイラによって、アクセスリスト A L cur の要素から取り除かれ、対応する空きポートカウンタ V P C における端子数を示すカウント値をデクリメントする。この後、コンパイラは、対応する出力端子のスイッチ状態表に調停に勝ったパケットの入力端子番号を記入し、調停に勝ったクラスタに処理済みのチェックを入れる。

【 0 0 6 2 】

次に、コンパイラは、調停に負けたパケットと、調停に勝ったパケットと同じあて先のクラスタを指定しているパケットとを、現在時刻のアクセスリスト A L cur から取り除き、次の時刻のアクセスリスト A L new に移動させる。コンパイラは、このような処理を、アクセスリスト A L cur のすべてのクラスタ A 0 ～ A 3 に対して行って処理済みチェックを入れるまで行う。アクセスリスト A L cur におけるすべてのクラスタ A 0 ～ A 3 に処理済みチェックを入れると、コンパイラ

は、アクセスリスト AL_{cur} の全要素をアクセスリスト AL_{new} に移し、該アクセスリスト AL_{new} を AL_{cur} として上記一連の処理を各クラスタごとの要素がすべてなくなるまで行う。

【0063】

図14は、アクセスリスト AL 及び空きポートカウンタ VPC を使用したクロス網内のスケジューリング方法を示したフローチャートであり、図14を用いてクロス網内のスケジューリング処理の流れについてもう少し詳細に説明する。なお、図14の各フローで行われる処理は、特に明記しない限りコンパイラによって行われる。

図14において、まず最初に、クロス網内のレベル0のクラスタでアクセスリスト AL が空でないクラスタの集合 UCL の内、最も要素の少ないクラスタを現在注目しているクラスタ番号 CL_{cur} に設定する（ステップ $S41$ ）。なお、最も要素数が少ないクラスタが複数ある場合は、いずれか1つを選択してクラスタ番号 CL_{cur} に設定する。

【0064】

次に、クラスタ番号 CL_{cur} のクラスタにおけるアクセスリスト AL_{cur} の要素であるパケットを1つ選択し（ステップ $S42$ ）、該選択したパケットにおけるあて先クラスタの空きポートカウンタ VPC が0であるか否かを調べる（ステップ $S43$ ）。ステップ $S43$ で、空きポートカウンタ VPC が0である場合（YES）、選択したパケットを、パケット集合 Uts とアクセスリスト AL_{cur} から取り除き、次の時刻のパケット集合 $Uts+1$ にずらし、時刻 Ts 以降の時刻に発行される該パケットは、重複がなくなるまで後の時刻のパケット集合にずらされ（ステップ $S44$ ）、ステップ $S42$ に戻る。

【0065】

また、ステップ $S43$ で、空きポートカウンタ VPC が0でない場合（NO）、選択したパケットを、空いている出力端子があるスイッチエレメント $SE0 \sim SE3$ の内、最も番号の小さいスイッチエレメントの出力端子に割り当てる（ステップ $S45$ ）。次に、クラスタ集合 UCL において、アクセスリスト AL_{cur} で競合したパケットを有するクラスタの有無を調べ（ステップ $S46$ ）、競合し

たパケットを有するクラスタがある場合 (YES)、該競合したパケットをアクセスリスト AL_{cur}から取り除いてアクセスリスト AL_{new}に加え (ステップ S47)、ステップ S46に戻る。

【0066】

また、ステップ S46で、競合したパケットを有するクラスタがない場合 (NO)、対応する空きポートカウンタ VPCのカウント値を1つ減らし、現在注目しているクラスタ番号 CL_{cur}をクラスタ集合 UCLから削除する (ステップ S48)。次に、クラスタ集合 UCLが空集合でないか否かを調べ (ステップ S49)、空集合でない場合は (YES)、ステップ S41に戻る。また、ステップ S49で、空集合の場合は (NO)、アクセスリスト AL_{cur}の全パケットをアクセスリスト AL_{new}に移して該アクセスリスト AL_{new}を AL_{cur}とし、アクセスリストが空でないクラスタをクラスタ集合 UCLの要素とする (ステップ S50)。次に、クラスタ集合 UCLが空集合か否かを調べ (ステップ S51)、空集合の場合は (YES)、本フローは終了し、空集合でない場合は (NO)、ステップ S41に戻る。

【0067】

このようなコンパイラによる処理を、具体的な例を用いて説明する。図15は、アクセスリスト AL_{cur}の初期状態の例を示した図であり、図16は、空きポートカウンタ VPCの初期状態の例を示した図である。図15及び図16で示した場合を例にして説明する。

まず、コンパイラは、アクセスリスト AL_{cur}における要素数の最も少ないクラスタ A1におけるあて先がクラスタ A2であるパケットを処理し、アクセスリスト AL_{cur}から削除する。更に、コンパイラは、レベル1のエクスチェンジャをなすスイッチエレメント SE0の出力端子 #2を確保してスイッチ状態表に記録する。

【0068】

次に、コンパイラは、空きポートカウンタ VPCにおける出力端子 #2のカウント値を1減らし、アクセスリスト AL_{cur}のクラスタ A1に処理済みのチェックを入れる。一方、コンパイラは、空きポートカウンタ VPCにおける出力端子

2 のカウンタ値が 0 であることから、アクセスリスト A L cur において、クラスタ A 3 におけるあて先がクラスタ A 2 であるパケットを削除して、再転送するために次の時刻のアクセスリスト A L new に移す。

【 0 0 6 9 】

次に、コンパイラは、アクセスリスト A L cur において、要素数が次に少ないクラスタ A 3 におけるあて先がクラスタ A 0 であるパケットを、スイッチエレメント S E 0 の出力端子 # 0 を確保してスイッチ状態表に記録する。更に、コンパイラは、空きポートカウンタ V P C の出力端子 # 0 のカウンタ値を 1 減らし、アクセスリスト A L cur のクラスタ A 3 に処理済みのチェックを入れる。

【 0 0 7 0 】

同様に、コンパイラは、アクセスリスト A L cur において、クラスタ A 2 におけるあて先がクラスタ A 0 であるパケットを削除して、再転送するために次の時刻のアクセスリスト A L new に移す。この時点で、クラスタ A 0 及び A 2 の要素数が共に 2 つとなることから、コンパイラは、調停を行ってクラスタ A 0 を先に処理するものとする。コンパイラは、アクセスリスト A L cur におけるクラスタ A 0 のあて先がクラスタ A 1 であるパケットを、スイッチエレメント S E 0 の出力端子 # 1 に割り当ててスイッチ状態表に記録し、アクセスリスト A L cur から削除する。

【 0 0 7 1 】

更に、コンパイラは、空きポートカウンタ V P C における出力端子 # 1 のカウンタ値を 1 減らし、アクセスリスト A L cur のクラスタ A 0 に処理済みのチェックを入れる。この後、コンパイラは、アクセスリスト A L cur において、クラスタ A 2 におけるあて先がクラスタ A 1 であるパケットを削除して、再転送するために次の時刻のアクセスリスト A L new に移す。最後に、コンパイラは、処理済みのチェックが入っていないクラスタ A 2 におけるあて先がクラスタ A 3 であるパケットを同様にして処理する。

【 0 0 7 2 】

更に、コンパイラは、空きポートカウンタ V P C の出力端子 # 3 のカウンタ値を 1 減らし、アクセスリスト A L cur のクラスタ A 2 に処理済みのチェックを入

れてアクセスリストAL_{cur}に対する処理が終了する。図17は、クラスタA0～A3の各パケットを1つずつ処理した後のアクセスリストAL_{new}を示し、図18は、クラスタA0～A3の各パケットを1つずつ処理した後の空きポートカウンタVPCを示している。

【0073】

次に、コンパイラは、新たなアクセスリストAL_{cur}に対して、上記と同様の処理を行う。この際、パケットは、コンパイラによって、スイッチエレメントSE1の出力端子に割り当てられる点異なる。このため、アクセスリストAL_{new}がAL_{cur}になるごとに、コンパイラによって割り当てられるスイッチエレメントが1つずつずれていくことになる。最終的に、コンパイラによってスケジューリングが施されたパケットの経路は、図19の矢印のようになる。

【0074】

なお、上記コンパイラによるスケジューリングでは、1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合に調停が行われ、出力端子を確保することができなかったパケットは次の時刻のスイッチ状態表で該出力端子を要求する場合を例にして説明したが、これは一例であり、出力端子を確保することができなかったパケットが、その前の時刻のスイッチ状態表といったように他の時刻のスイッチ状態表で所望の出力端子を要求するようにしてもよい。

【0075】

このように、本第1の実施の形態におけるマルチプロセッサシステム装置は、各プロセッサエレメント間を、階層構造を有する多段結合網で接続し、該多段結合網を構成する各スイッチに対して、あらかじめコンパイラによって静的にスケジューリングを行い、階層構造を有する多段結合網を無衝突でエミュレーションするようにした。このことから、パケットの衝突時に動的に行っていたパケットの待ち合わせを、すべてコンパイル時に管理することができるため、パケットの動的な待ち合わせに必要なFIFO等のハードウェアを大幅に削減することができる。更に、マルチプロセッサシステム装置において無同期実行を可能にすることができるため、同期させるためのハードウェアのオーバヘッドを削減すること

ができ、並列処理の効率を向上させることができる。

【 0 0 7 6 】

また、階層構造を有する多段結合網の基本網にクロス網を使用して1つのクロス網内でパケット転送を行う場合、レベル1のエクステンジャのスイッチエレメントに対するスケジューリングを行った際、調停に負けたパケットをレベル1のエクステンジャの他のスイッチエレメントにおける空きのスイッチを使用して転送するようにしてもよい。このようにすることによって、パケット転送効率を向上させることができる。

【 0 0 7 7 】

第2の実施の形態。

上記第1の実施の形態では、クロス網間を接続するための2段目のレベル1のエクステンジャ、すなわち図6における各クロス網内のそれぞれのスイッチエレメントSE0～SE3にすべてのパケットが集中する構造であるため、ホットスポット (hot spot) が形成されて性能が著しく低下する場合があった。このことから、上位階層から下位階層へ配送するダウンストリーム用のスイッチとして、レベル1のコンセントレータを付加するようにしてもよく、このようにしたものを本発明の第2の実施の形態とする。なお、本第2の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図、及びプロセッサエレメントの構成例を示した概略のブロック図は、図1及び図2と同様であるので省略する。

【 0 0 7 8 】

図20及び図21は、本発明の第2の実施の形態における階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。図20は、各クロス網と拡張ネットワークとのアップリンク結合を示しており、図21は、各クロス網と拡張ネットワークとのダウンリンク結合を示している。なお、図20及び図21では、図6と同じものは同じ符号で示しており、ここではその説明を省略すると共に図6との相違点のみ説明する。また、図20及び図21においても、4を基数、すなわち4つのプロセッサエレメントPE0～PE3を有する4つのクラスタA0～A3を備える4つのクラスタD0～D3で構成された階

層構造クラスタリングを実現した場合を例にして示し、プロセッサエレメントは省略して示している。

【 0 0 7 9 】

図 2 0 及び図 2 1 における図 6 との相違点は、レベル 1 のエクスチェンジャにおける、上位階層ネットワークへのパケット配送機能（アップストリーム）と、下位階層ネットワークへのパケット配送機能（ダウンストリーム）を 2 つに分け、上位階層から下位階層へパケットを配送するダウンストリーム用のスイッチとしてスイッチングエレメント SC b 0 ～ SC b 3 からなるレベル 1 のコンセントラータを付加すると共に、下位階層から上位階層へパケットを配送するアップストリームは、第 1 の実施の形態と同様に各クラスタ D 0 ～ D 3 におけるそれぞれのスイッチングエレメント SE 0 ～ SE 3 からなるレベル 1 のエクスチェンジャによって行われる。

【 0 0 8 0 】

あるプロセッサエレメント内のプロセッサ PU が、他のプロセッサエレメント PE のプロセッサ PU とデータの授受を行う場合、相手のプロセッサエレメント PE のメモリ ME にデータを書き込むことによって通信を行う。メモリ ME にデータが書き込まれたプロセッサエレメントのプロセッサ PU は、メモリ ME に書き込まれたデータを読み出すことによりデータ通信が成立する。

【 0 0 8 1 】

以下、プロセッサエレメント間のデータ通信の流れについて図 2 2 を用いて説明する。

図 2 2 において、プロセッサエレメント PE a からプロセッサエレメント PE b にデータ転送する場合を例にして説明する。まず、プロセッサエレメント PE a において、プロセッサ PU a からネットワークインタフェース NI a にアドレス及び転送データが送られる。

【 0 0 8 2 】

次に、ネットワークインタフェース NI a は、入力されたアドレスに基づいてパケットを生成し、階層構造の多段結合網 MIN にパケットを投入する。該投入されたパケットは、階層構造の多段結合網 MIN を介して、プロセッサエレメン

ト P E b のネットワークインタフェース N I b に入力され、該ネットワークインタフェース N I b は、入力されたパケットを解体してメモリ M E b に書き込む。プロセッサ P U b は、メモリ M E b に書き込まれたデータを読み出してデータ通信が完了する。

【 0 0 8 3 】

ここで、レベル 1 のエクスチェンジャに送り出されたパケットが、同一のクロス網内で処理されるパケット、すなわちデスティネーションが同一クロス網内である場合について図 3 を用いて説明する。

図 3 において、プロセッサエレメントからパケットが、1 段目のディストリビュータに送り出され、該 1 段目のディストリビュータでスイッチングされて 2 段目のレベル 1 のエクスチェンジャに送られる。レベル 1 のエクスチェンジャに送り出されたパケットは、該レベル 1 のエクスチェンジャによって、最終段のレベル 0 のコンセントレータにスイッチングされて送り出される。

【 0 0 8 4 】

更に、レベル 0 のコンセントレータに送り出されたパケットは、レベル 0 のコンセントレータで適切にスイッチングされて、デスティネーションのプロセッサエレメントに送り出されて階層構造の多段結合網 M I N でのデータ通信が完了する。なお、あて先のプロセッサエレメントに送り出されたパケットは、図 2 2 で説明したように、プロセッサエレメントのメモリに格納される。

【 0 0 8 5 】

次に、レベル 1 のエクスチェンジャに送り出されたパケットが、他のクロス網で処理されるパケット、すなわちデスティネーションが他のクロス網である場合について図 2 3 を用いて説明する。なお、図 2 3 では、プロセッサエレメント P E a からプロセッサエレメント P E b にデータ転送する場合を例にして説明する。

レベル 1 のエクスチェンジャをなすスイッチエレメント S E 1 に送り出されたパケットは、該スイッチエレメント S E 1 によって、拡張段への出力端子にスイッチングされ、同一レベルのクラスタ内に入るまで上位階層のエクスチェンジャ、この場合レベル 2 のエクスチェンジャをなすスイッチエレメント S E a 1 に送

られる。

【0086】

同一クラスタ内に入ったパケットは、適切な出力にスイッチングされて階層を下る。例えば図23の場合、スイッチエレメントSE a 1に送られたパケットは、スイッチエレメントSE a 1によって、ダウンストリームのレベル1のコンセントレータをなすスイッチエレメントSC b 1に転送される。スイッチエレメントSC b 1に転送されたパケットは、スイッチエレメントSC b 1で適切にスイッチングされて、デスティネーションのプロセッサエレメントであるプロセッサエレメントPE bに送り出される。このようにして、階層構造の多段結合網MINでのデータ通信が完了する。このような構成において、階層構造を有する多段結合網の静的スケジューリング方法は上記第1の実施の形態と同様であるのでその説明を省略する。

【0087】

このように、本第2実施の形態におけるマルチプロセッサシステム装置は、上位階層から下位階層へ配送するダウンストリーム用のスイッチとして、スイッチングエレメントSC b 0～SC b 3からなるレベル1のコンセントレータを付加し、下位階層から上位階層へパケットを配送するアップストリームは、スイッチングエレメントSE 0～SE 3からなるレベル1のエキスチェンジャによって行うようにした。このことから、クロス網間を接続するためのレベル1のエキスチェンジャにすべてのパケットが集中しないようにしてホットスポットの形成を防止し、マルチプロセッサシステム装置の性能向上を図ることができる。

【0088】

【発明の効果】

上記の説明から明らかなように、本発明のマルチプロセッサシステム装置によれば、各プロセッサエレメント及び多段結合網を、所定の数をもとにした階層構造にクラスタリングすると共に、各時刻ごとに生成された多段結合網における各スイッチの状態を示すスイッチ状態表を用いてあらかじめ静的にスケジューリングされたスケジュールに基づいて、プロセッサエレメント間のパケット転送を行うようにした。このことから、マルチプロセッサシステム装置において無同期実

行を可能にすることができるため、同期させるためのハードウェアのオーバーヘッドを削減することができ、並列処理の効率を向上させることができる。

【 0 0 8 9 】

また、上記階層構造を有する多段結合網は、下位階層から上位階層へパケット転送を行うアップストリーム用の結合網と、上位階層から下位階層へパケット転送を行うダウンストリーム用の結合網とをそれぞれ備えるようにした。このことから、クロス網間を接続するためのエクステンジャをなす結合網にすべてのパケットが集中しないようにしてホットスポットの形成を防止し、マルチプロセッサシステム装置の性能向上を図ることができる。

【 0 0 9 0 】

具体的には、上記スイッチ状態表を、各スイッチごとの、出力端子を保持しているパケットの情報と、該出力端子を要求しているパケットの情報と、該出力端子の状態を示した情報とで構成した。このことから、各プロセッサエレメント及び多段結合網からなる大規模なシステムに対する静的なスケジューリングを容易に行うことができる。

【 0 0 9 1 】

また、多段結合網における1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、他の時刻のスイッチ状態表で該出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにした。このことから、一般的な同時アクセスパターンに対して、無衝突のパケット転送を実現することができる。

【 0 0 9 2 】

一方、上記多段結合網がクロス網の場合、1つのクロス網内のパケット転送時に、多段結合網における1つのスイッチの出力端子を同一時刻で複数のパケットが要求した場合、所定の方法で調停が行われ、該出力端子を保持できなかったパケットは、パケットの要求がない他のスイッチの出力端子を要求するようにしてスケジューリングされたスケジュールにしたがって、プロセッサエレメント間のパケット転送を行うようにしてもよい。このようにすることによって、パケット

転送効率を向上させることができ、マルチプロセッサシステム装置の性能向上を図ることができる。

【 0 0 9 3 】

具体的には、各パケットに対するスケジューリングをあらかじめコンパイラによって行うようにした。このことから、パケットの衝突時に動的に行っていたパケットの待ち合わせを、すべてコンパイル時に管理することができるため、パケットの動的な待ち合わせに必要な F I F O 等のハードウェアを大幅に削減することができプロセッサ間での無同期実行を行うためのネットワーク環境を整えることができる。

【図面の簡単な説明】

【図 1】 本発明の第 1 の実施の形態におけるマルチプロセッサシステム装置の例を示した概略のブロック図である。

【図 2】 プロセッサエレメントの構成例を示した概略のブロック図である。

【図 3】 クロス網の例を示した図である。

【図 4】 クロス網内におけるレベル 0 のネットワークの例を示した図である。

【図 5】 クロス網内におけるレベル 1 のネットワークの例を示した図である。

【図 6】 階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。

【図 7】 スイッチ状態表の例を示した図である。

【図 8】 スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図 9】 スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図 1 0】 スイッチ状態表を使用した静的スケジューリング方法を示したフローチャートである。

【図 1 1】 調停を行う前のスイッチ状態表の例を示した図である。

【図 1 2】 調停を行った後のスイッチ状態表の例を示した図である。

【図 1 3】 クロス網の例を示した図である。

【図 1 4】 アクセスリスト A L 及び空きポートカウンタ V P C を使用したクロス網内のスケジューリング方法を示したフローチャートである。

【図 1 5】 アクセスリスト A L cur の初期状態の例を示した図である。

【図 1 6】 空きポートカウンタ V P C の初期状態の例を示した図である。

【図 1 7】 各パケットを 1 つずつ処理した後のアクセスリスト A L new の例を示した図である。

【図 1 8】 各パケットを 1 つずつ処理した後の空きポートカウンタ V P C の例を示した図である。

【図 1 9】 スケジューリング後の各パケットの経路を示した図である。

【図 2 0】 本発明の第 2 の実施の形態における階層構造クラスタリングを実現したマルチプロセッサシステム装置の例を示した図である。

【図 2 1】 図 2 0 のマルチプロセッサシステム装置における各クロス網と拡張ネットワークとのダウンリンク結合例を示した図である。

【図 2 2】 プロセッサエレメント間のデータ通信の流れの例を示した図である。

【図 2 3】 図 2 0 のマルチプロセッサシステム装置 1 a におけるプロセッサエレメント間のパケット転送の流れの例を示した図である。

【符号の説明】

- 1, 1 a マルチプロセッサシステム装置
- P E, P E 0 ~ P E m プロセッサエレメント
- A 0 ~ A n レベル 0 のクラスタ
- B 0 ~ B n レベル 0 の結合網
- C 0 ~ C x レベル 1 の結合網
- D 0 ~ D x レベル 1 のクラスタ
- E 0 レベル 2 の結合網
- P U, P U a, P U b プロセッサ
- M E, M E a, M E b メモリ

NI, NI a, NI b ネットワークインタフェース

SC0～SC3 コンセントレータ（レベル0）のスイッチエレメント

SD0～SD3 ディストリビュータ（レベル0）のスイッチエレメント

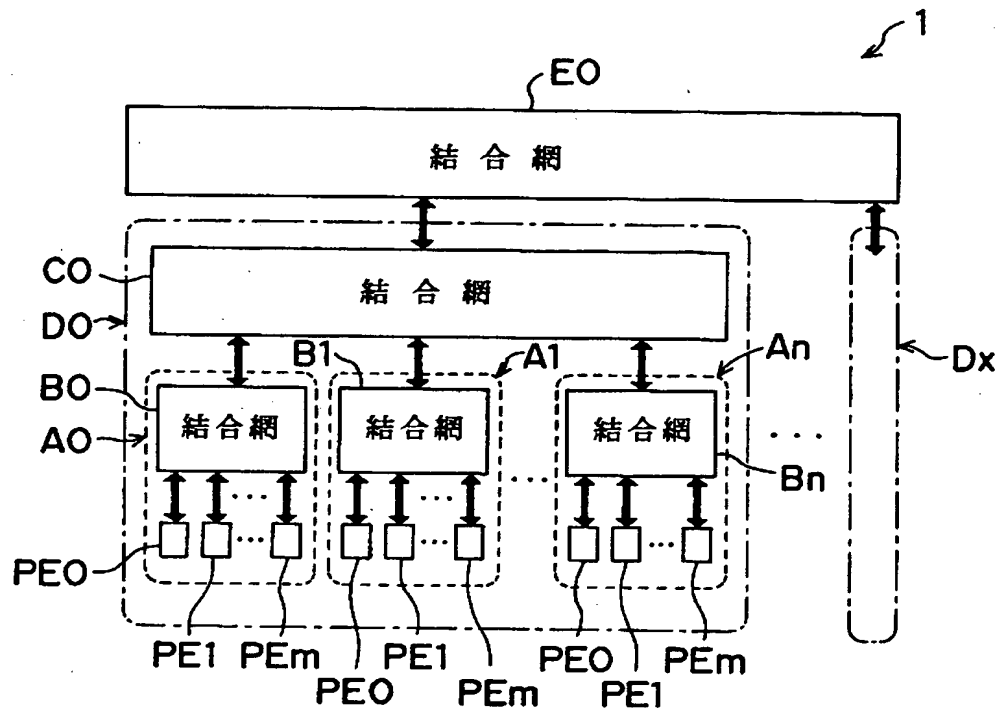
SE0～SE3 エクスチェンジャ（レベル0, 1）のスイッチエレメント

SE a 0～SE a 3 エクスチェンジャ（レベル2）のスイッチエレメント

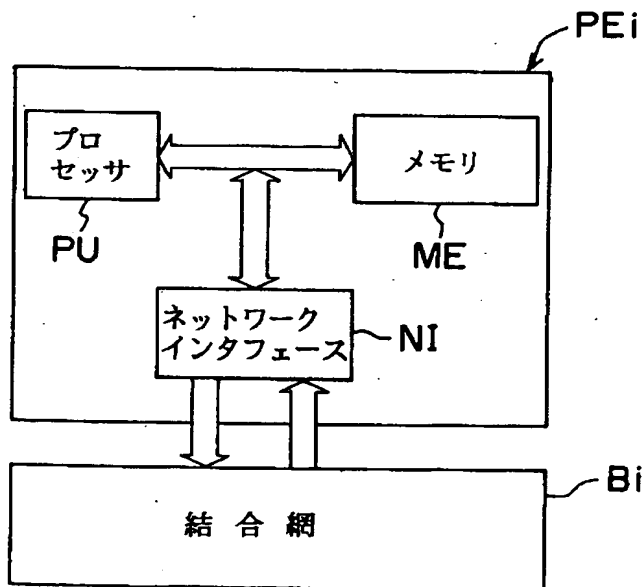
SC b 0～SC b 3 コンセントレータ（レベル1）のスイッチエレメント

【書類名】 図面

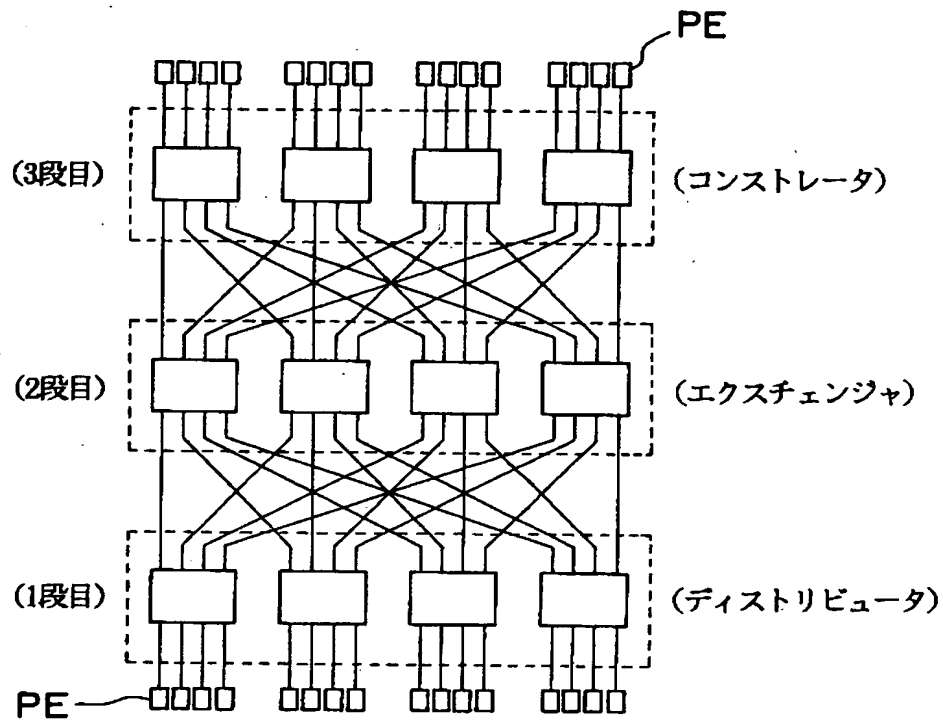
【図 1】



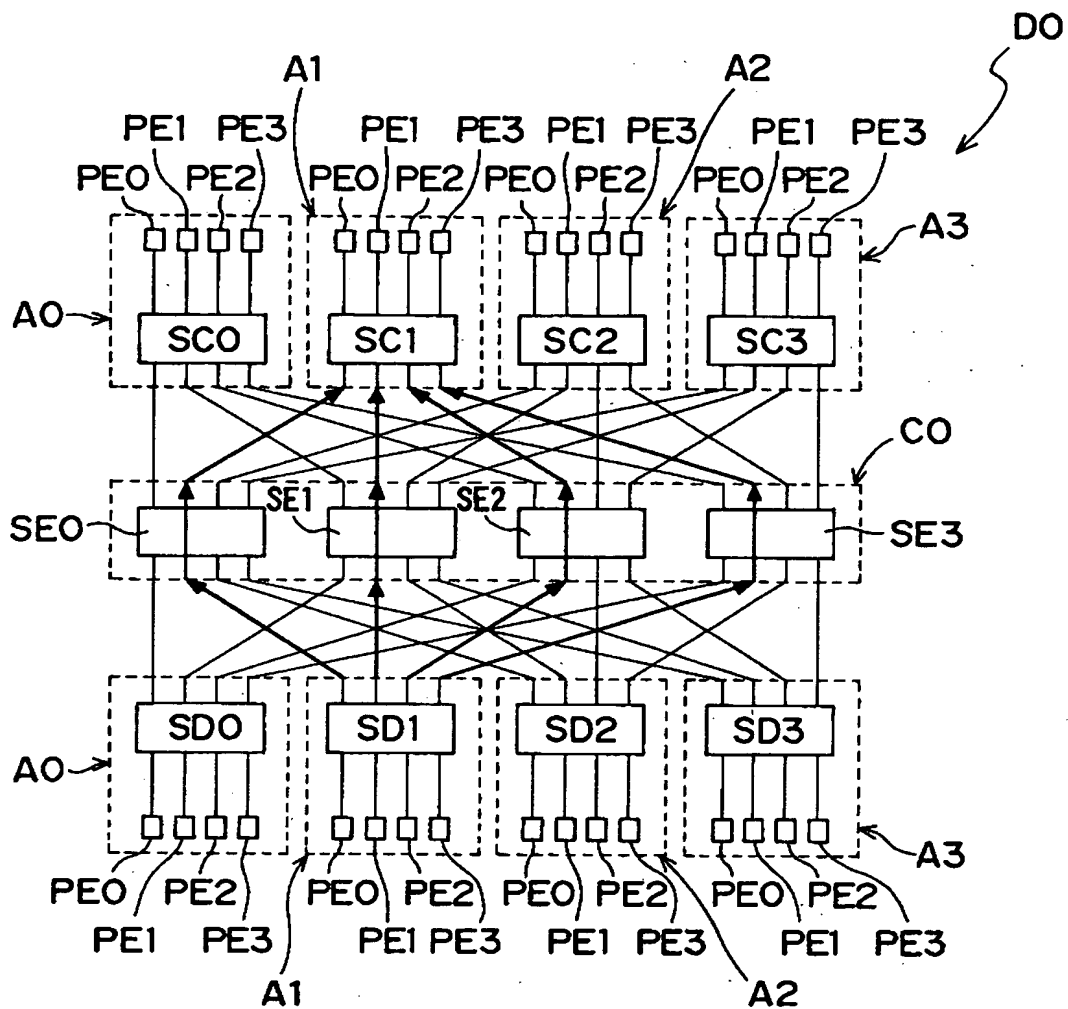
【図 2】



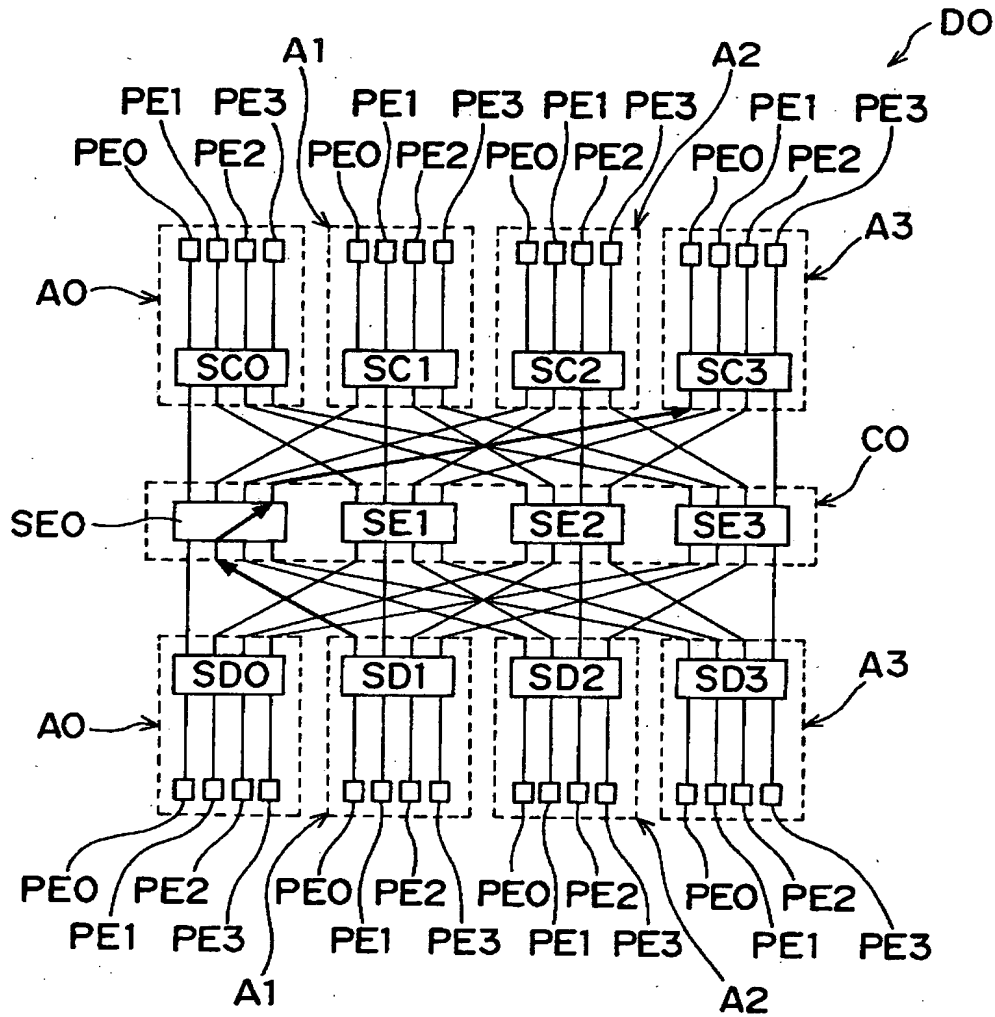
【図 3】



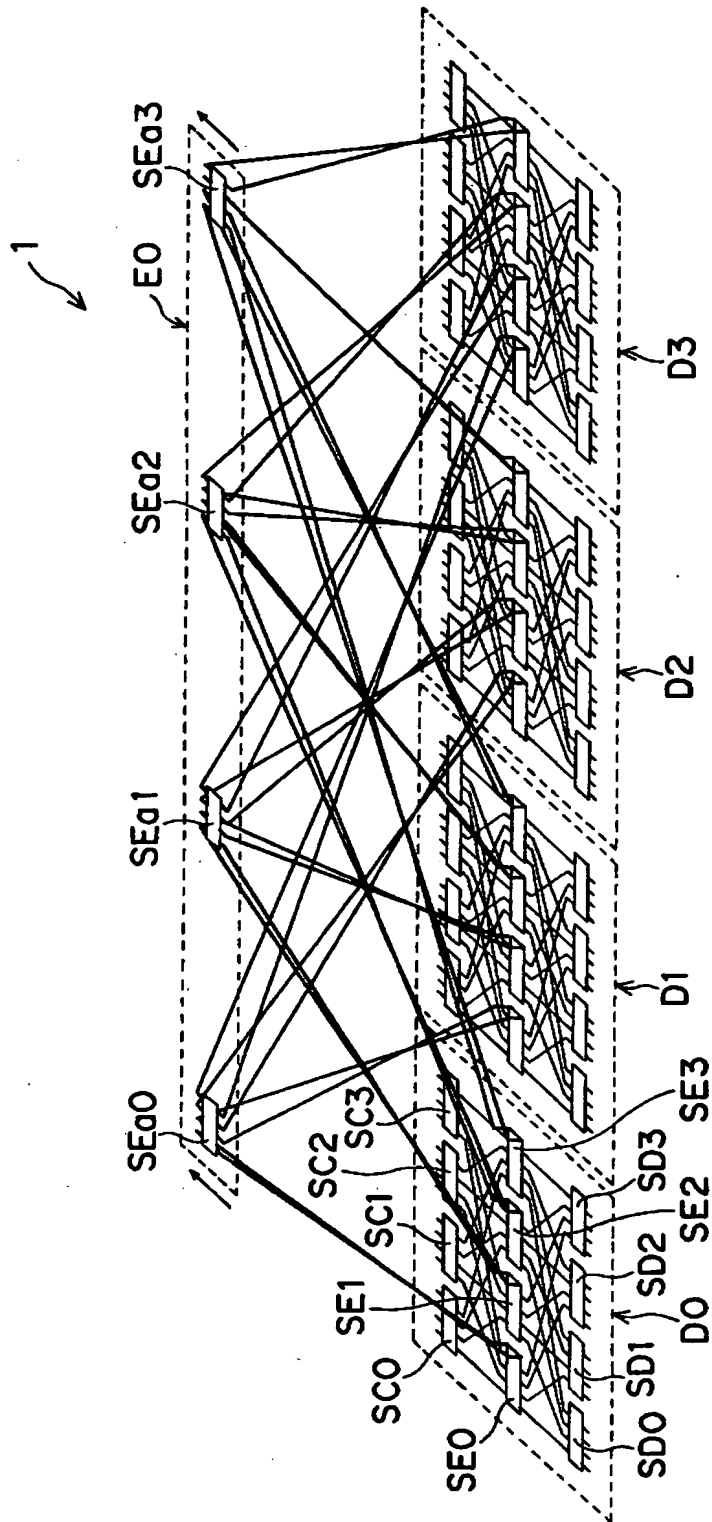
【図4】



【図5】



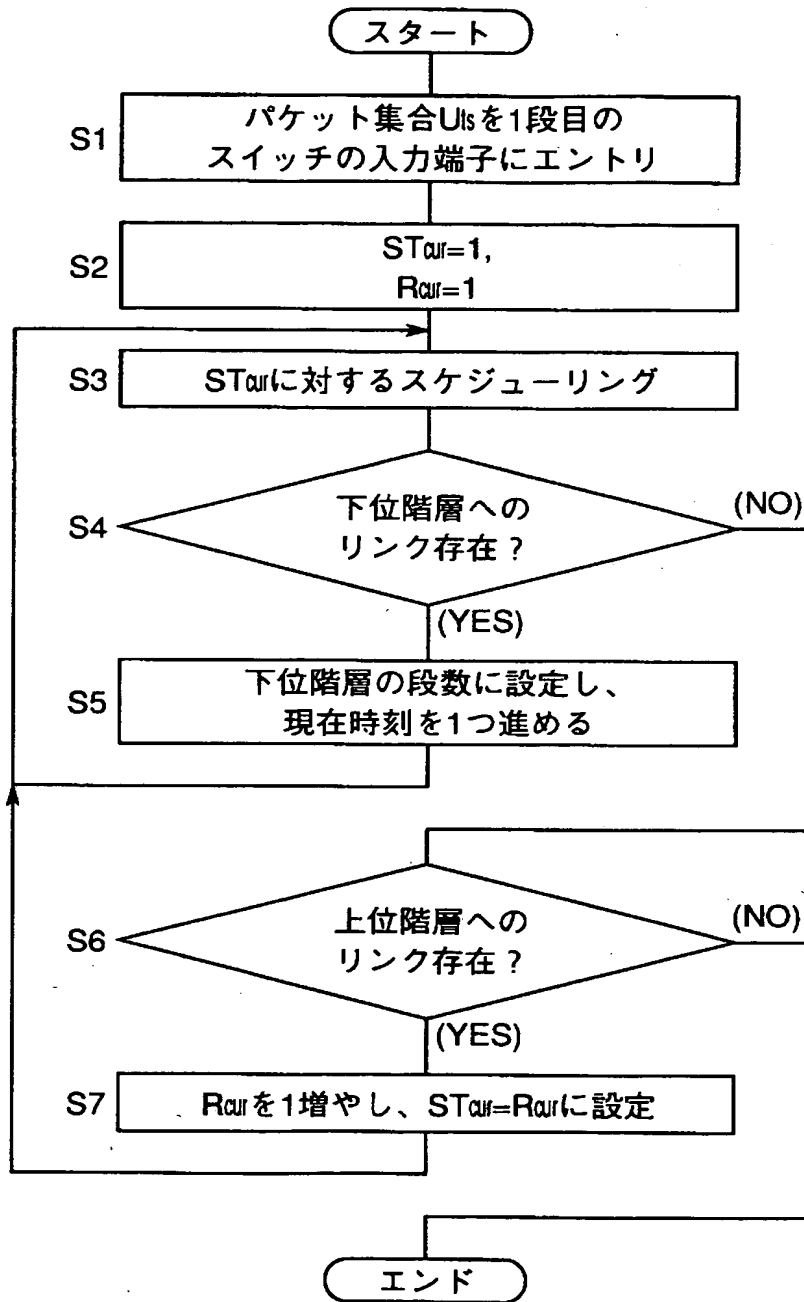
【図 6】



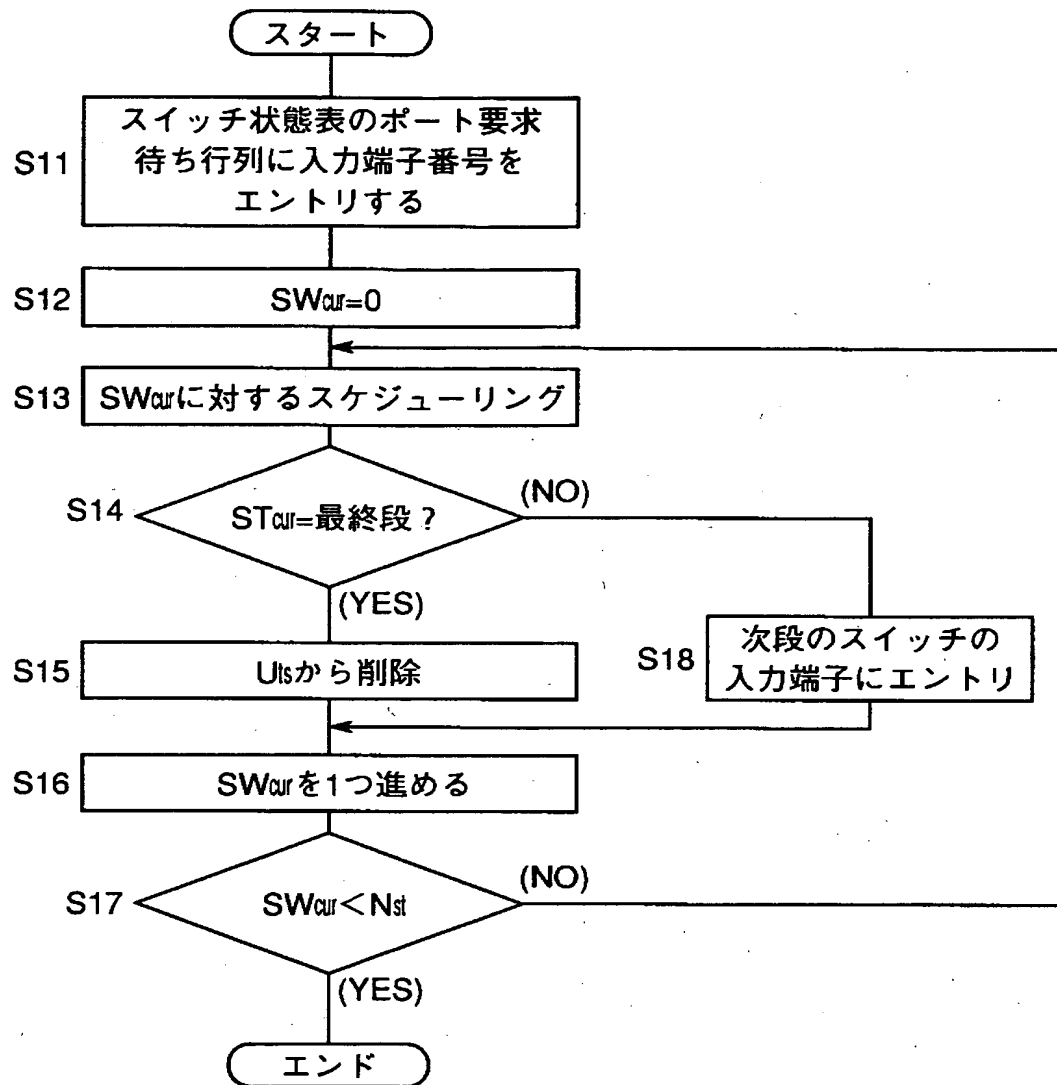
【図 7】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
#0	157843	#3	2	-	ホールド
#1	157843	-	0	#0, #2	リリース
#2	157843	-	0	#1	リリース
#3	157843	-	0	-	リリース

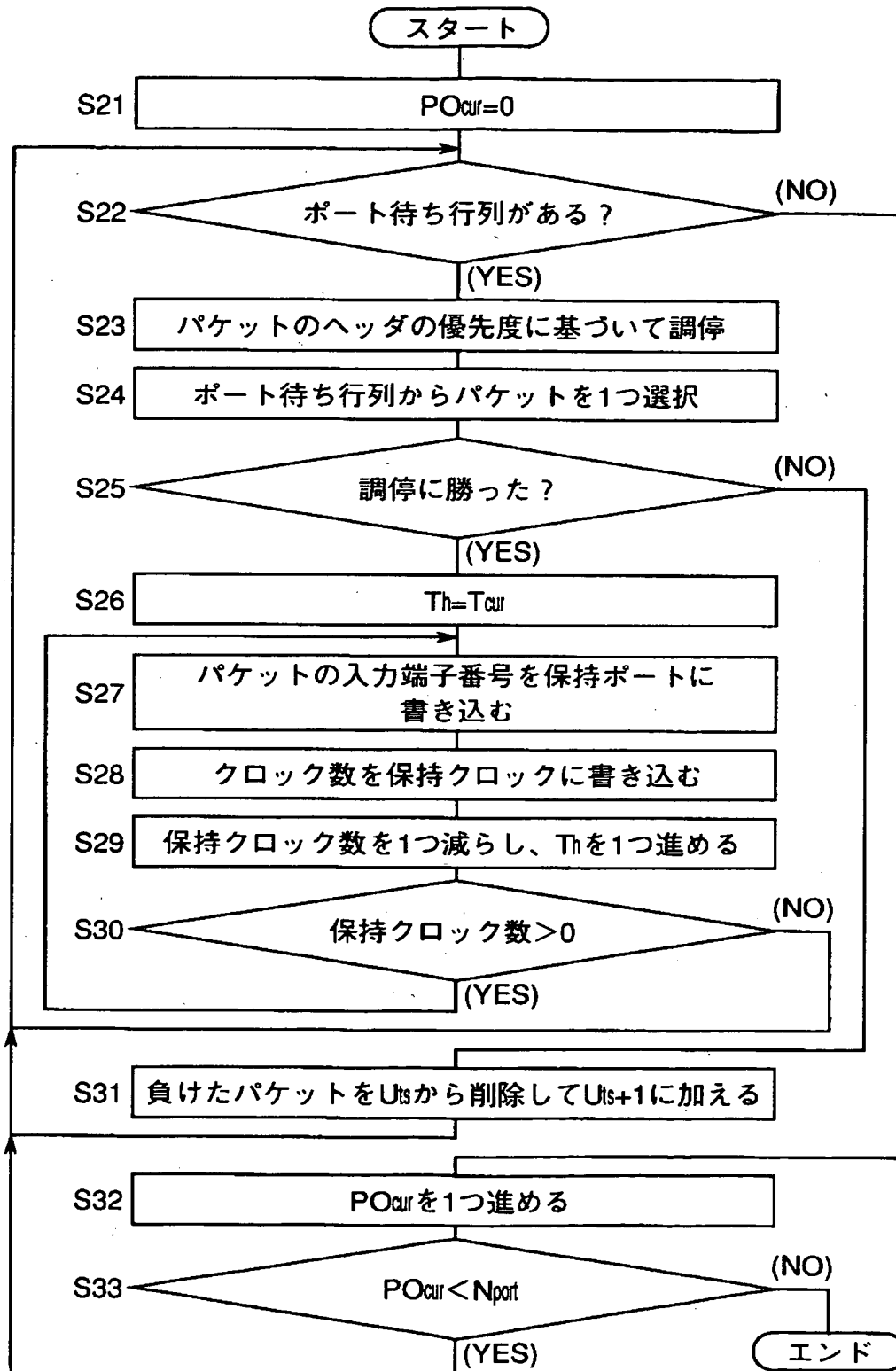
【図 8】



【図 9】



【図10】



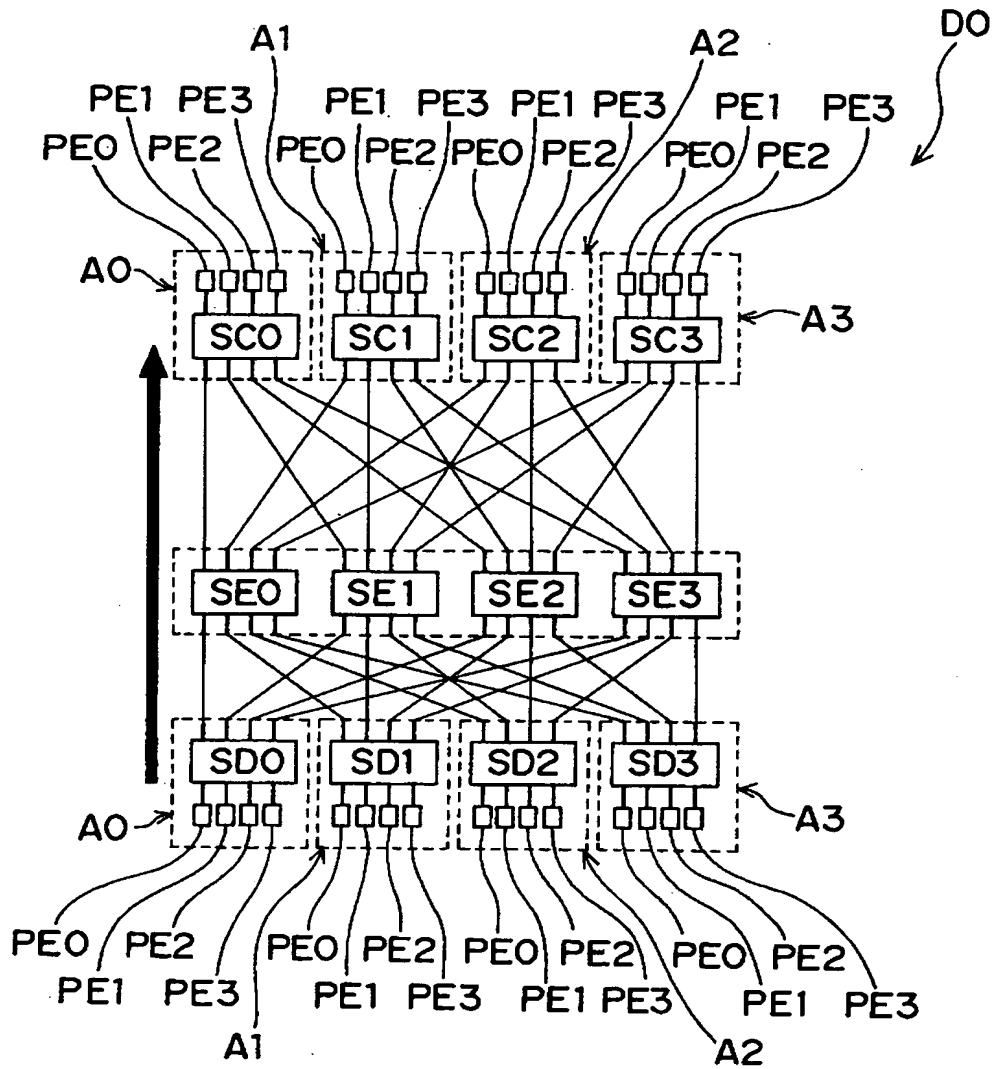
【図 1 1】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
#0	15000	-	0	-	リリース
#1	15000	-	0	#0, #1	リリース
#2	15000	#4	2	#3	ホールド
#3	15000	-	0	-	リリース
#4	15000	-	0	#2	リリース

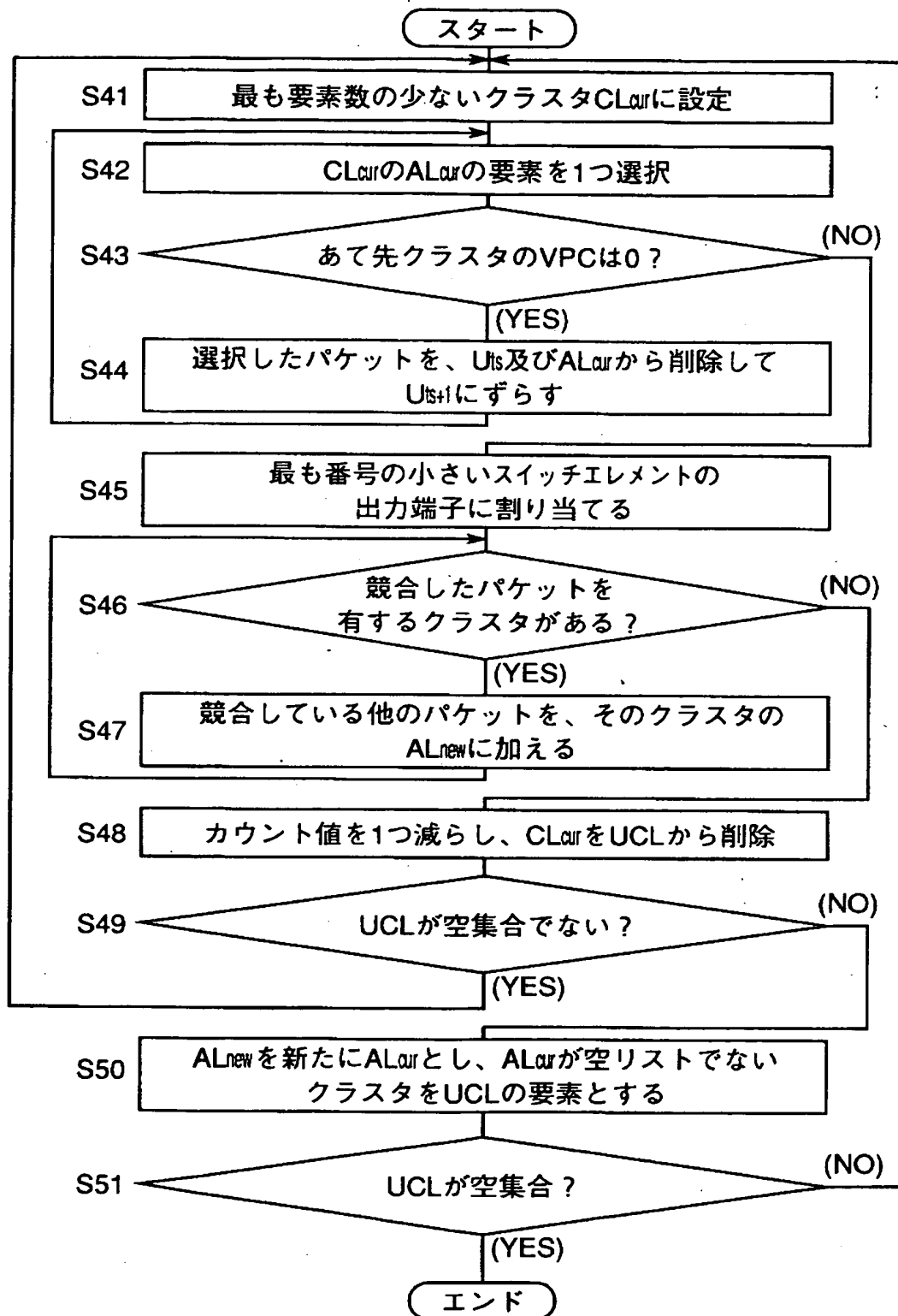
【図 1 2】

出力端子番号	現在時刻	保持ポート	保持クロック	ポート要求待ち行列	状態
#0	15000	-	0	-	リリース
#1	15000	#1	1	-	ホールド
#2	15000	#4	2	-	ホールド
#3	15000	-	0	-	リリース
#4	15000	#2	1	-	ホールド

【図13】



【図 14】



【図 15】

クラス番号	あて先のクラス番号
A0	A1,A3
A1	A2
A2	A0,A1,A3
A3	A0,A2

【図 16】

出力端子番号	#0	#1	#2	#3
カウント値	2	3	1	2

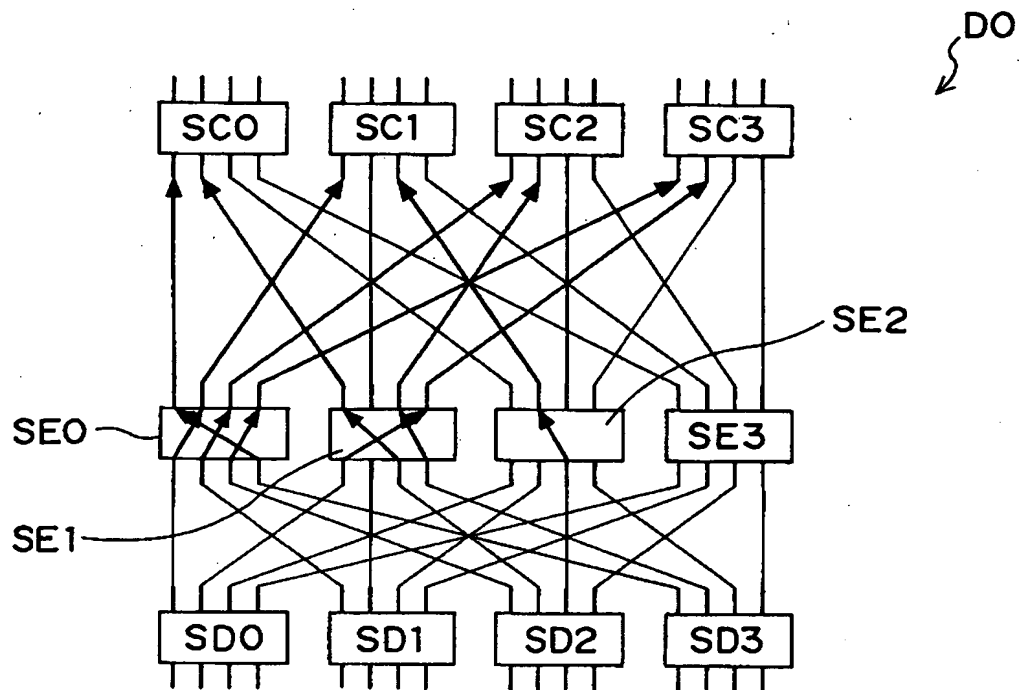
【図 17】

クラス番号	あて先のクラス番号
A0	A3
A1	—
A2	A0,A1
A3	A2

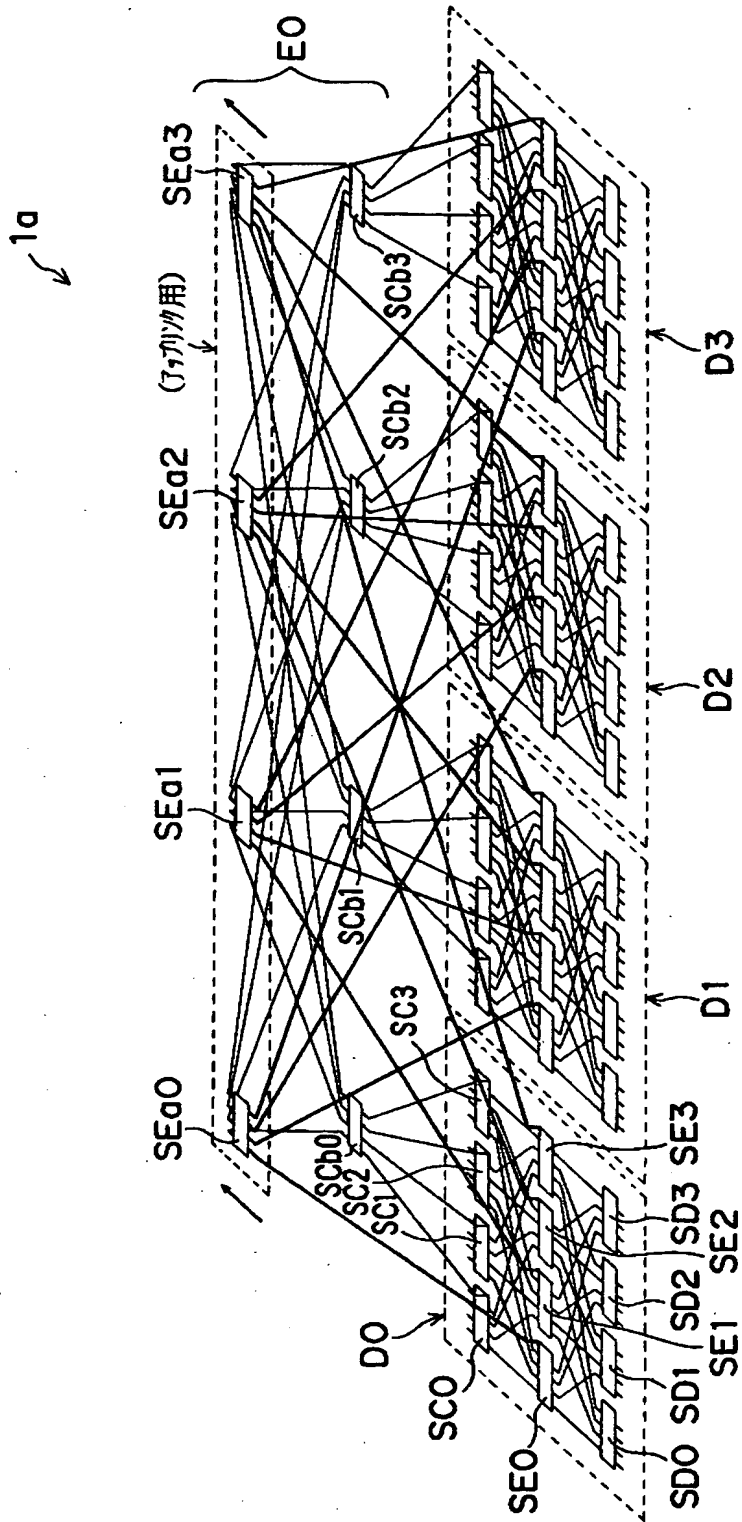
【図 18】

出力端子番号	#0	#1	#2	#3
カウント値	1	2	0	1

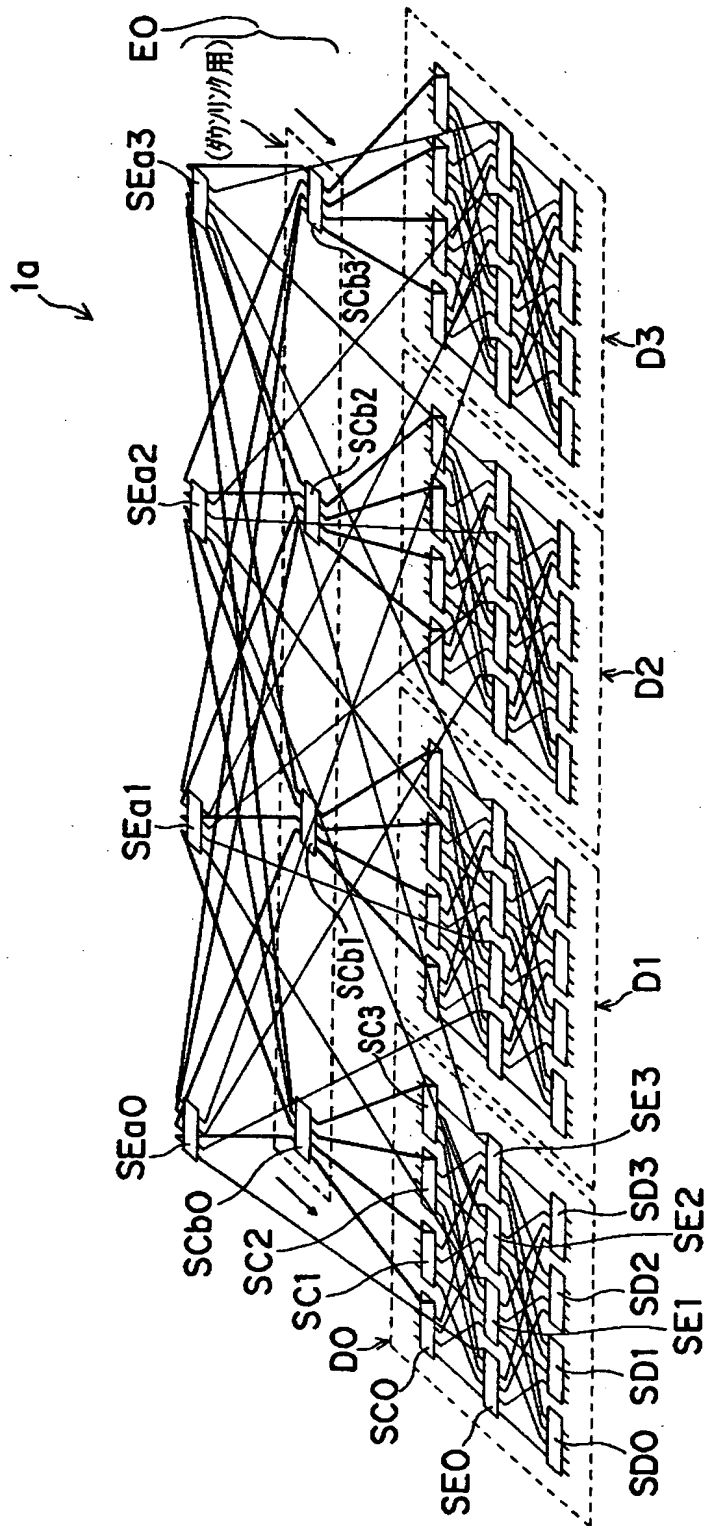
【図 1 9】



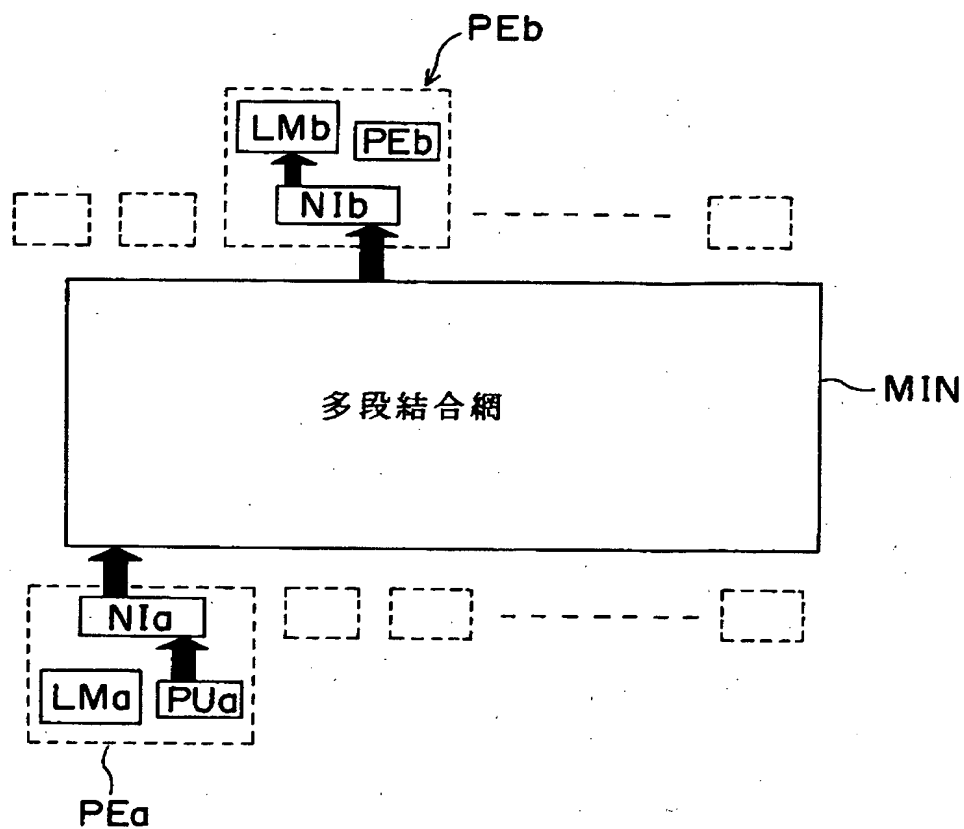
【図20】



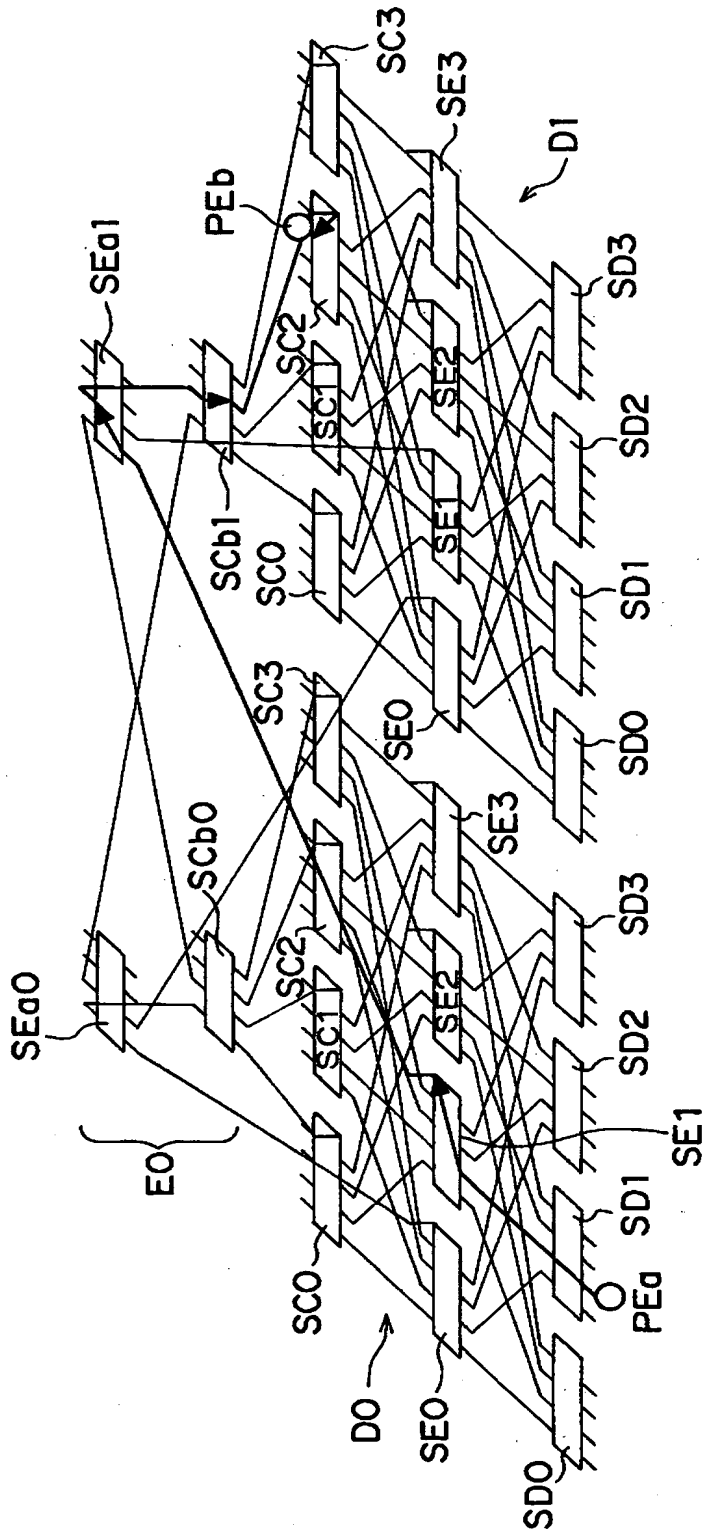
【図21】



【図 2 2】



【図23】



【書類名】 要約書

【要約】

【課題】 コンパイラが容易に静的スケジューリングを行うことができ、一般的な同時アクセスパターンに対して無衝突なパケット転送を実現することができるマルチプロセッサシステム装置を得る。

【解決手段】 各プロセッサエレメント間を、階層構造の多段結合網で接続し、該多段結合網を構成する各スイッチエレメントに対して、あらかじめコンパイラによって静的にスケジューリングを行い、階層構造の多段結合網を無衝突でエミュレーションするようにした。更に、階層構造の多段結合網の基本網にクロス網を使用して1つのクロス網内でパケット転送を行う場合、レベル1のエクステンションのスイッチエレメントSE0～SE3に対するスケジューリングを行った際、調停に負けたパケットをスイッチエレメントSE0～SE3の他のスイッチエレメントにおける空きスイッチを使用して転送するようにした。

【選択図】 図6

出 願 人 履 歴 情 報

識別番号 [396023993]

1. 変更年月日 1996年10月28日
[変更理由] 新規登録
住 所 東京都港区新橋6丁目16番10号
氏 名 株式会社半導体理工学研究センター
2. 変更年月日 2001年 3月23日
[変更理由] 住所変更
住 所 神奈川県横浜市港北区新横浜3丁目17番地2 友泉新横浜ビ
ル6階
氏 名 株式会社半導体理工学研究センター